TR-792-8-283
NOVEMBER 1967

ANALYTICAL RESEARCH IN GUIDANCE THEORY

FINAL TECHNICAL REPORT

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
ELECTRONICS RESEARCH CENTER
Guidance Laboratory

UNDER CONTRACT NAS12-500

NORTRONICS – HUNTSVILLE
NORTHROP CORPORATION
6025 TECHNOLOGY DRIVE, P O BOX 1484 HUNTSVILLE, ALABAMA 35805
TELEPHONE 837-0580

113

ANALYTICAL RESEARCH IN GUIDANCE THEORY

FINAL TECHNICAL REPORT

November 1967

Nortronics-Huntsville Technical Report No. 351

by

J. F. Andrus
R. C. Armstrong
F. C. Heilscher
D. R. Kilpatrick
S. S. Hu
R. Silber
J. W. Newkirk
J. Zien
M. L. Thompson

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
ELECTRONICS RESEARCH CENTER
GUIDANCE LABORATORY

Under Contract NAS12-500

Reviewed & Approved by:

W. G. Eaton

W. G. Eaton, Director, Space Sciences Section

Steve S. Hu

Dr. Steve S. Hu, Technical Director

NORTRONICS-HUNTSVILLE
HUNTSVILLE, ALABAMA

FOREWORD

This report summarizes studies performed under Contract NAS12-500 with the Guidance Laboratory, NASA Electronics Research Center, Cambridge, Massachusetts. The period of performance was from 15 September 1966 through 31 October 1967. Technical director for the contract was Mr. W. E. Miner of the Electronics Research Center Guidance Laboratory.

Nortronics personnel who contributed to this study and this report are listed on the cover page. The writing and organization of this report was done by M. L. Thompson.

Computer time needed during the course of this study was obtained from the Marshall Space Flight Center's Computation Laboratory under the NASA Resource Sharing Program.

SUMMARY

Described in this report are studies completed under Contract NAS12-500 with the NASA Electronics Research Center, Cambridge, Massachusetts.

These studies are concerned with series solutions of two-point boundary condition problems that result from applying the calculus of variations to optimal guidance problems. The major objective was to use these solutions to devise accurate, analytical approximations to guidance functions for optimal ascent to orbit. This work is essentially an extension of that completed under Contract NASW-1165, which is described in reference 1.

An outstanding problem associated with these studies is the need to perform algebraic manipulations of lengthy and complicated formulas. In particular the multiplication of symbolic, polynomial-like expressions occurs frequently. An integral part of the work was to devise new techniques or apply existing ones to automate symbolic mathematical operations on digital computers.

In addition, research in the calculus of variations was carried out by Mr. Robert Silber. The result of these efforts was a modification of the Bolza problem to include end-orbits governed by n-body attractions. This work is available in reference 2 and is not discussed at length in this report.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

GENERAL STATEMENT OF GUIDANCE PROBLEM AND APPROACH TO SOLUTION

In this section, only a cursory description of the problem treated and the approach to solution is given. A detailed description has been given in reference 1.

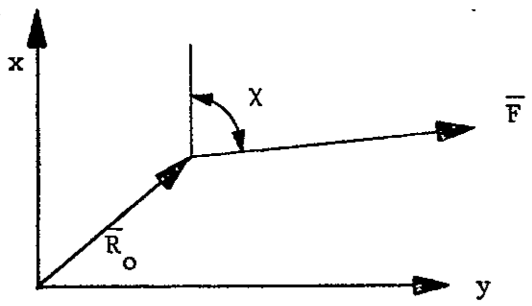
1.1 DEFINITIONS

We make the following assumptions and definitions regarding the vehicle and its performance:

- Constant thrust magnitude, F, and mass flow rate, $\dot{m}$.
- Rocket is represented by a point mass, m.
- Single-stage rocket with continuous thrusting to orbit.
- Spherical, non-rotating, airless central body of attraction.
- Trajectory and orbit are coplanar.
- The optimum trajectory is the one that requires a minimum time from initial state to terminal orbit.

1.2 DIFFERENTIAL EQUATIONS

The coordinate system used is shown in the following figure. The equations of motion are written in this system



$$\ddot{x} = \frac{F}{m} \sin \chi - g_x$$

$$\ddot{y} = \frac{F}{m} \cos \chi - g_y$$

Application of the calculus of variations yields an additional system of differential equations. These equations may be coupled with the preceding equations to yield; in vector form:

$$\dot{\overline{V}} = \frac{F}{m} \frac{\overline{\lambda}}{|\overline{\lambda}|} - \beta_1 \overline{R}$$

$$\ddot{\overline{\lambda}} = \beta_1 \overline{\lambda} + (\overline{\lambda} \cdot \overline{R}) \gamma \overline{R} \qquad (1)$$

where

$$\beta_1 = \left(\frac{GM}{R^3}\right)$$

$$\gamma = -3 \left(\frac{GM}{R^5}\right)$$

The second of equations (1) is usually called the Euler-Lagrange equation, and the $\overline{\lambda}$ and $\dot{\overline{\lambda}}$ are called Lagrange multipliers.

1.3 BOUNDARY CONDITIONS

The initial conditions on the solutions of equations (1) are known with the exception of $\overline{\lambda}(t_o)$ and $\dot{\overline{\lambda}}(t_o)$. At the unspecified terminal time, t_f, it is necessary that the solution of equations (1) satisfy functions that generally are of the form

$$F_j(\overline{R}, \overline{V}, \overline{\lambda}, \dot{\overline{\lambda}}) = 0 \quad (j = 1,\ldots,4) \qquad (2)$$

Some of these F_j are the transversality conditions and some are relations that describe the desired terminal orbit.

1.4 APPROACH TO SOLUTION

We first note that there are as many boundary conditions to be satisfied as there are unknown initial conditions. Because of the homogeneity of the Euler-Lagrange equation, one of the multipliers may be set to an arbitrary value at the initial point. The same thing can be accomplished by letting $|\overline{\lambda}| = 1$ at $t = t_o$. This additional condition will also introduce simplifications in subsequent developments. For brevity we will refer to this relation as the "scaling condition".

We now have a total of five conditions on the solutions of equations (1) and also five unknowns. In order to make use of the F_j we expand them in Taylor series about the interval $(t_f - t_o)$, to express them approximately at the initial point. This yields a system of equations of the form:

$$\sum_{n=0}^{k} \frac{1}{n!} \frac{d^n}{dt^n} (F_j) \Bigg|_{t_0} (t_f - t_0)^n \approx 0 \qquad (3)$$

Repeated substitution of equations(1) and their derivatives in the series coefficients of equations (3) is used to reduce equations (3) to a system of algebraic equations in the Lagrange multipliers. Equations (3), together with the scaling condition, then implicitly define values for the multipliers and the flight time, $(t_f - t_o)$, which we define as Δt.

Numerical verification of the validity of this approach for ascent to circular orbits has been demonstrated previously for numerous trajectories (see reference 1). Usually the error in the solution for the multipliers is 1 percent or less for series of order four or greater.

Actually, the solution for the multipliers was not carried out simultaneously with the solution for Δt. Instead, one of the series expansions was solved for Δt explicitly by a procedure known as "series inversion" and the result was substituted into the other expansions. The reason for this is discussed in the next section.

Now, with Δt eliminated from the equations we have four algebraic equations in the four unknown Lagrange multipliers. These equations can be solved for values of the multipliers at the initial point when values of the state variables and constants are known. It can also be shown that a simple relationship exists between the multipliers and the steering angle, χ for an optimal trajectory [cf. equations (1)]. Thus, a solution for the multipliers is a solution to the optimal guidance problem.

Subsequent sections of this report discuss the application of this approach to three different missions; optimal ascent to circular orbit, optimal ascent to rendezvous with a circular orbiting target satellite, and optimal ascent to elliptical orbit.

In order to test or verify the validity of the analytical developments in this study, comparison is made with numerically computed, optimum trajectories. These trajectories are called "nominal". Values for the state variables and constants are taken from these nominal trajectories and used in the numerical evaluation of formulas. Comparisons are then made between nominal values for the multipliers and flight time and those values predicted by the formulas.

SECTION II
OPTIMAL ASCENT TO CIRCULAR ORBIT

2.1 INTRODUCTION

In this guidance problem we are concerned with the problem of steering a vehicle along an optimum path to insertion into a circular orbit of specified altitude.

The terminal functions to be satisfied are:

$$V_f^2 - V_{co}^2 = 0$$

$$R_f^2 - R_{co}^2 = 0 \tag{4}$$

$$\overline{R}_f \cdot \overline{V}_f = 0$$

$$\lambda_1 v - \lambda_2 u + \lambda_3 y - \lambda_4 x = 0$$

It can be shown that the last of equations (4), the transversality condition, is for this problem, an integral of the differential equations and is valid at t_o. It can be applied directly without being expanded in a series, as can the scaling condition, $|\overline{\lambda}| = 1$, by definition at $t = t_o$.

The first of equations (4) will be used to illustrate the expansion in a Taylor series.

$$V_f^2 - V_{co}^2 = 0 \approx (V^2 - V_{co}^2)_o + 2(\overline{V}\cdot\dot{\overline{V}})_o\ (\Delta t)$$
$$+ (\overline{V}\cdot\ddot{\overline{V}} + \dot{\overline{V}}\cdot\dot{\overline{V}})_o\ (\Delta t)^2 + \frac{1}{3!}\ (\overline{V}\cdot\dddot{\overline{V}} + 3\dot{\overline{V}}\cdot\ddot{\overline{V}})_o)\ (\Delta t)^3 + \cdots \qquad (5)$$

where $\Delta t = (t_f - t_o)$ and the subscript "o" indicates a value at $t = t_o$.

After the remaining functions are expanded in a similar fashion, the derivatives of the differential equations can be substituted into the series' coefficients to reduce equations (4) to a system of five nonlinear algebraic equations in the four Lagrange multipliers and Δt. (For reference, these derivatives are listed in Appendix A.)

At this point we could solve these five simultaneous equations. However, we choose to eliminate Δt from the equations by a technique called "series inversion". One of the series is inverted to obtain an analytical expression for Δt and this expression is then substituted for Δt in the other expansions. There are two reasons for this approach. First, one of the series may, for a given order, implicitly define more accurate values for Δt than the other series. In fact this was shown to be the case in reference 1. Second, it may be desired to calculate approximate Δt values independently of the multipliers during the guidance process.

2.2 SOLUTION FOR MULTIPLIERS

The solution of the systems of algebraic equations for the four Lagrange multipliers was discussed at length in reference 1. Details to be found there will not be discussed here.

Of major concern during this study was the possible simplification of analytical derivations. It has been verified previously that systems of equations can be derived that define quite accurate solutions for the multipliers. More accuracy can be attained by using higher order series. Therefore, primary attention was directed toward analytical solutions of the equations that define the multipliers, elimination of insignificant terms, and determining approximations associated with certain classes of missions. For convenience and simplicity, the systems of equations derived from third-order series were used in the search for analytical solutions.

Two methods for solution of the equations for the multipliers were examined. The method of "Successive Substitutions" was considered as a means of getting an explicit formula for the multipliers as functions of the local state variables and mission constants. At first it seemed that this technique could be easily applied. But in order to obtain the third step approximations, about 100,000 terms would be involved. A few algebraic simplifications such as combining terms were possible but did not cause any significant reduction in numbers of terms. It was decided that, if used, this method should be applied after the coefficients in the equations were numerically evaluated.

The second method considered was to invert the system of equations in the multipliers. This would lead to less complicated formulas for the solution but not much is known about the practical question of convergence. Formulas for this inversion are derived in Appendix B. For illustrative purposes, the formulas are for two series in two variables. The inversion technique should converge rapidly if estimates of the multipliers are available. By restricting the problem to a class of trajectories it may be possible to expand the equations in the multipliers about a "standard estimate" associated with a class of trajectories. One would then have a system of equations in which the variables are corrections to the estimates. If the corrections are relatively smaller than the estimates, there should be many opportunities for simplification and approximation.

An approach analogous to the above (but in one variable) was taken in the solution for time-to-go or Δt, described in the following subsection. There, a great amount of simplification was possible.

2.3 SOLUTION FOR FLIGHT TIME (Δt)

Of the three possible expansions to be inverted for Δt, the one describing the terminal velocity condition, equation (5) was chosen. The reasons for this choice were: (1) A third-order series implicitly defines accurate Δt values. (2) Velocity, for an optimal trajectory, should be a monotone increasing function of time and the possibility of a singularity is avoided.

We rewrite equation (5) as

$$Z = A_1 \Delta t + A_2 \Delta t^2 + A_3 \Delta t^3 \tag{6}$$

where

$$Z = \left(V_{co}^2 - V_o^2\right)/2\bar{V} \cdot \dot{\bar{V}}$$

$$A_1 = 1$$

$$A_2 = (\bar{V} \cdot \ddot{\bar{V}} + \dot{\bar{V}} \cdot \dot{\bar{V}})/2\bar{V} \cdot \dot{\bar{V}}$$

$$A_3 = (\dot{\bar{V}} \cdot \dddot{\bar{V}} + 3\dot{\bar{V}} \cdot \ddot{\bar{V}})/6\bar{V} \cdot \dot{\bar{V}}.$$

Equation (6) can be inverted to obtain an approximation for Δt

$$\Delta t \simeq B_1 Z + B_2 Z^2 + B_3 Z^3 + \ldots \tag{7}$$

It often happens that the inverse series, equation (7), requires an unwieldy number of terms to approximate a root of equation (6); and the coefficients of the inverse series become progressively more complicated. The number of terms needed in the inverse series can be significantly reduced by expanding equation (6), not about the origin but about an estimate of t_f. Define

$$\Delta t = \tau + dt,$$

where τ is an estimate and dt a correction to τ. Substitution into equation (6) yields

$$Z' = A_1' dt + A_2' dt^2 + A_3' dt^3 \tag{8}$$

Equation (8) can now be solved for the correction by a series of inversion.

There are several means of estimating Δt. One way is to take only the linear terms of equation (6) and have

$$\Delta t \simeq Z = \tau \qquad (A_1 = 1).$$

Another way is to use the "rocket equation" which is fairly accurate for short-arc flights:

$$\tau = \frac{m}{\dot{m}} (1 - e^A),$$

where

$$A = -(V_{co} - V_o)/V_{exhaust}.$$

A third method of estimating time is to assume that it is known from a previous in-flight computation, say, five seconds ago, or that the expected flight time is known from preflight planning. This latter possibility is quite attractive, because the Δt estimation formula is rather tolerant of "poor" estimates.

Considerable simplification is possible in equation (8). It is only necessary, for most estimates, to invert the series to one term; i.e.,

$$Z' = A_1' \, dt$$

$$dt = \frac{Z'}{A_1'},$$

where

$$Z' = Z - \left(A_1\tau + A_2\tau^2 + A_3\tau^3\right)$$

$$A_1' = A_1 + 2\tau A_2 + 3\tau^2 A_3$$

Within the A_1' coefficient it was possible to neglect the term $3\tau^2 A_3$ and frequently possible to neglect the term $2\tau A_2$. However, to assure accuracy over a reasonable range of τ values, the latter term was retained with a further approximation. The formula for A_2 was approximated as

$$A_2 \approx \dot{\overline{V}} \cdot \dot{\overline{V}}/2\overline{V} \cdot \dot{\overline{V}} = c$$

The expression for Δt in simplified form is

$$\Delta t = \tau + \frac{Z - \left(\tau + A_2\tau^2 + A_3\tau^3\right)}{1 + 2C}$$

or

$$\Delta t = \tau + dt_A, \qquad (9)$$

In cases where it might be necessary to invert equation (8) to higher orders than the first, the A_i' coefficients can be replaced with the A_i coefficients of equation (6) with no significant loss of accuracy.

The terms Z and C both involve terms with $\dot{\bar{V}}$ in their denominators. This leads to formulas that involve λ_1 and λ_2 in the denominators when the equations of motion are substituted. To circumvent this difficulty, another approximation was made. It is assumed that the components of $\dot{\bar{V}}$, in denominators only, would be treated as measured accelerations rather than the accelerations associated with the optimal trajectory. This approximation was tested in equation (9) by using accelerations obtained by perturbations on the steering angle around a nominal, optimum value. The effect on Δt values was relatively insignificant as Table 2-1 indicates.

Table 2-1. EFFECT ON Δt VALUES

% ERROR IN χ	ERROR IN Δt (SEC)
1	0.45
2	0.85
3	1.5
4	1.6
5	1.66
-1	0.04
-2	0.5
-3	1.1
-4	2.0
-5	2.7

No further simplifications in equation (9) appear to be possible. For missions different from those represented by the nominal data used, the approximations may be invalid.

If equation (9) is written in terms of the multipliers after substitution of the differential equations, there is a total of 19 terms. (These are tabulated in Appendix C.) This is an improvement over results previously obtained (reference 1) where the Δt formula had 50 terms when written in terms of the multipliers.

The great difficulty in eliminating Δt from the equations is substitution of a formula such as equation (9) into the other series where it must be squared, cubed, etc. But note that if τ is a good estimate for Δt, then dt_A is relatively smaller than τ. We would then expect the following approximations to the powers of Δt to be sufficiently accurate.

$$(\Delta t)^2 \approx \tau^2 + 2\tau dt_A$$
$$(\Delta t)^3 \approx \tau^3 + 3\tau^2 dt_A$$
etc.

In fact, for a comfortable range of τ, these formulas approximate the powers of Δt rather well. Representative data is shown in Figures 2-1 through 2-3.

It should be noted that Δt <u>and</u> its powers can be approximated by expressions that involve only 19 terms in the multipliers. Further, the algebraic form of the approximations is the same for the different powers. Only the coefficients need be slightly altered by a constant multiplicative factor.

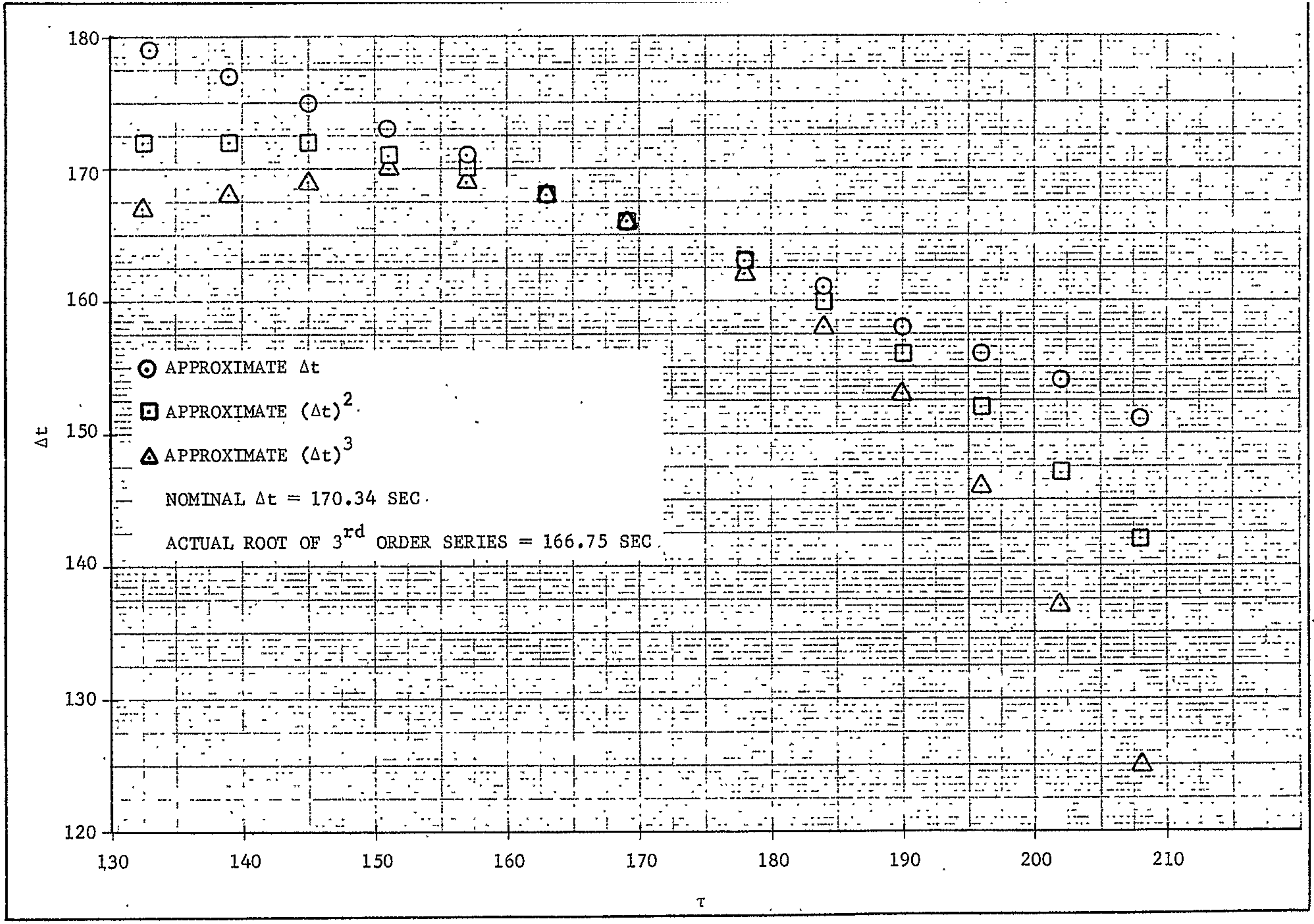
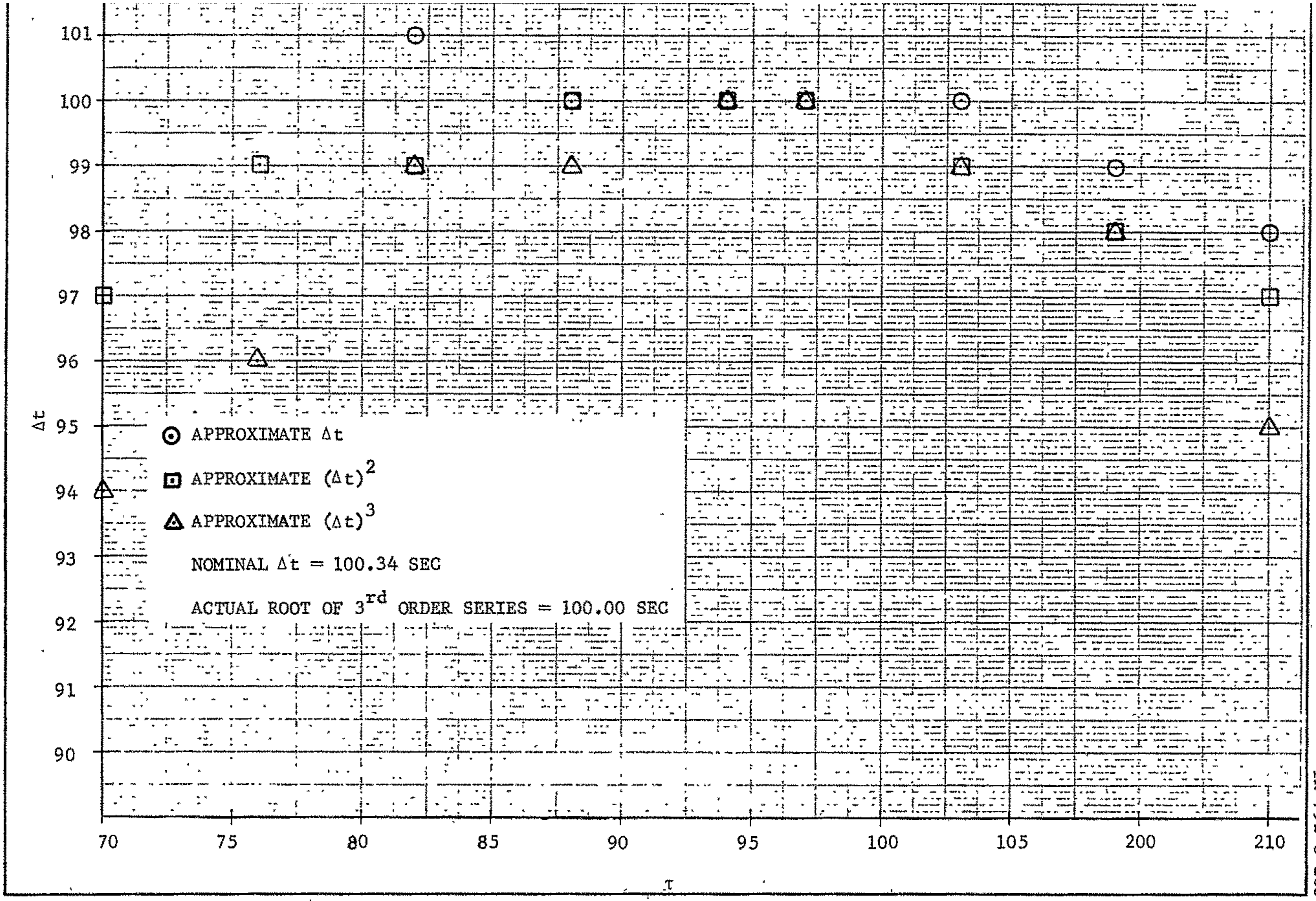


Figure 2-1. APPROXIMATIONS TO Δt FOR DIFFERENT VALUES OF τ



TR-792-8-283

Figure 2-2. APPROXIMATIONS TO Δt FOR DIFFERENT VALUES OF τ

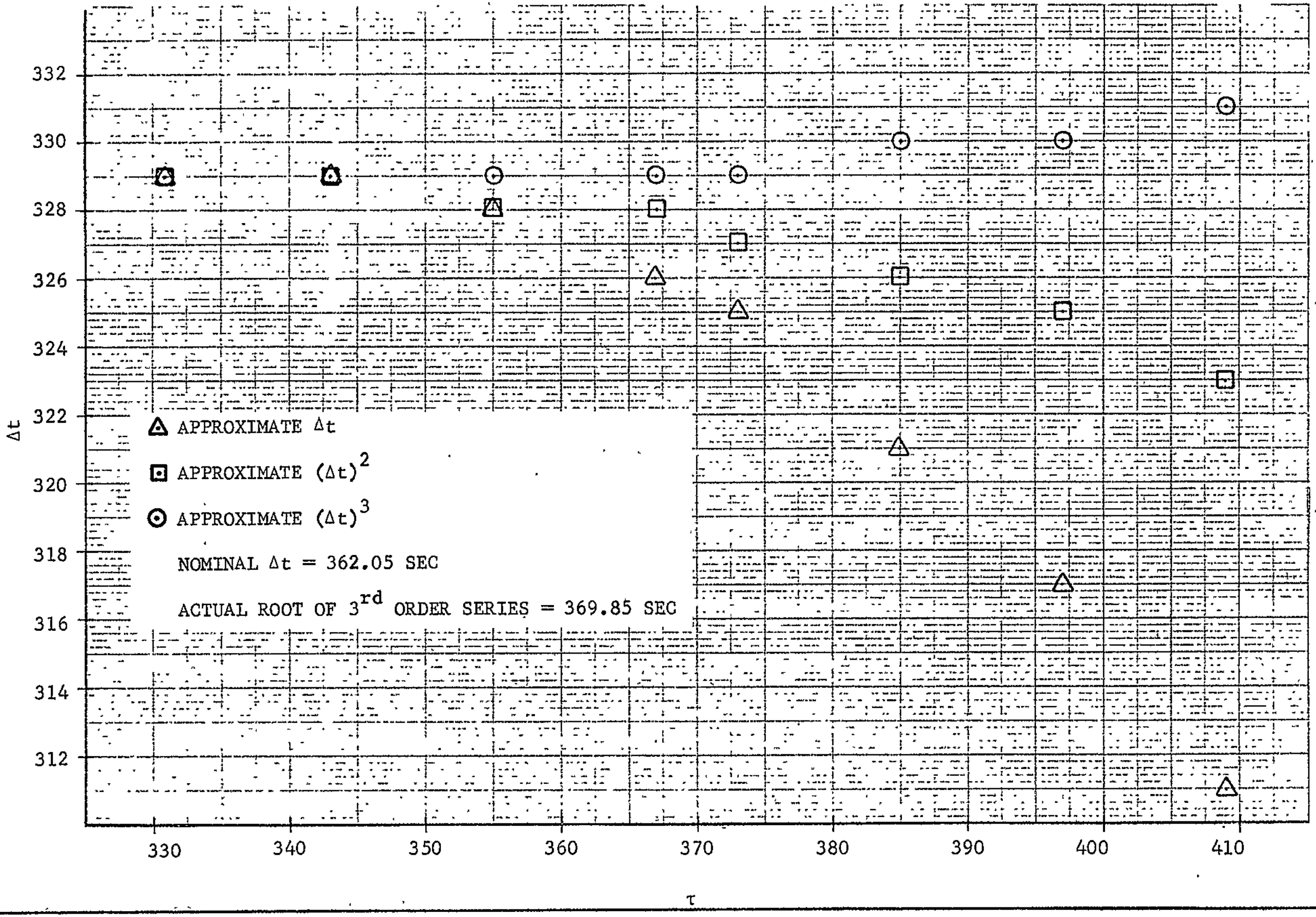


Figure 2-3. APPROXIMATIONS TO Δt FOR DIFFERENT VALUES OF τ

TR-792-8-283

The effect of these approximations on the solution for the multipliers was estimated without actually substituting the Δt formula. This was done by evaluating the approximations for Δt and its powers with nominal data, including the multipliers. The coefficients of the equations in the multipliers were evaluated with these Δt values instead of nominal Δt values, and then solved. The effect of the approximations for Δt was a small decrease in accuracy in the solution for the multipliers from that obtained when nominal Δt values were used. The difference between the two solutions ranged from less then one percent to about five percent in the worst case.

2.4 SUBSTITUTION OF Δt EXPRESSION INTO OTHER SERIES

As discussed previously, the expression for Δt is to be substituted into the other two series in order to have a system of four equations in the multipliers. From equation (3), it is evident that the coefficients of Δt in the series will generally be polynomial-like expressions in the multipliers. We are thus faced with the rather formidable problem of substituting one multivariate polynomial into another. The problem can be represented as determining the right-hand side of the equation:

$$P_o + P_1Q_1 + P_2Q_2 + P_3Q_3 + \dots = S,$$

where the P's and Q's are multivariable polynomials. The right-hand side, S, should be simplified in the sense that like powers of the variables and like coefficients are collected.

Note that the problem is considerably simplified if the approximations to Δt, previously described, are used. In that event, each of the Q's is essentially the same as far as algebraic manipulation is concerned. Only the coefficients associated with each Q need be distinguished. The problem is still not practically manageable if done by hand.

The Northrop MULPO program, developed under Contract NAS12-500, was used to perform these algebraic manipulations. (See Section V for a more detailed description.) The analytical approximations for Δt and its powers were substituted into third-order expansions of the terminal functions that specify cutoff radius and path angle. The results are shown in Tables 2-2 and 2-3. (See subsection 5.3 for a key to reading Tables 2-2 and 2-3.) There is a total of 100 different combinations of lambda exponents in one expansion (radius) and 176 in the other (path angle). As an estimate of the reduction in numbers of terms generated, the Δt expression without truncation of powers was substituted and the result simplified by MULPO. The resulting series had over 9000 terms.

2.5 SIMPLIFICATION OF COEFFICIENTS

Attempts were made to eliminate terms in the system of equations that define the initial values of the multipliers. (The formula for Δt had not been substituted.) This problem was discussed previously in reference 1. Essentially no simplifications other than those described in reference 1 were achieved.

One approach that was tried was to write the differential equations' derivatives in vector form and then attempt to drop numerically insignificant terms. (See Appendix A.) This approach was less fruitful than working with components of the vectors as was done previously.

It was noted that simplifications that are made according to the behavior on one trajectory are also valid for neighboring trajectories that have different initial values. Thus, simplifications already found are valid over a class or field of trajectories.

After the analytical approximations for Δt and its powers are substituted, as shown in Tables 2-2 and 2-3, there may be possibilities for further simplifications. The system of equations in the multipliers obtained after substitution of Δt has not been coded for solution because of lack of available time. However, Northrop is sponsoring an effort to code a computer program to solve these equations numerically in order to determine their validity. A part of this effort would also involve the elimination of terms that do not contribute significantly to the solution.

Table 2-2. SERIES FOR RADIUS CONDITION

							L1	L2	L3	L4	L5	L6
							0	0	0	0		
A^1	0*	0*	0*	0*	1*							
B^1	0*	0*	0*	0*	1*	0*	0*	0*	0*	1*		
C^1	0*	0*	0*	0*	1*	0*	0*	0*	0*	1*		
D^1	0*	0*	0*	0*	1*	0*	0*	0*	0*	1*		
							L1	L2	L3	L4	L5	L6
							0	0	0	1		
D^1	0*	0*	0*	1*	1*	0*	0*	0*	0*	1*		
							L1	L2	L3	L4	L5	L6
							0	0	1	0		
D^1	0*	0*	1*	0*	1*	0*	0*	0*	0*	1*		
							L1	L2	L3	L4	L5	L6
							0	1	0	0		
B^1	0*	0*	0*	0*	1*	0*	1*	0*	0*	1*		
C^1	0*	0*	0*	0*	1*	0*	1*	0*	0*	1*		
C^1	0*	1*	0*	0*	1*	0*	0*	0*	0*	1*		
D^1	0*	0*	0*	0*	1*	0*	1*	0*	0*	1*		
D^1	0*	1*	0*	0*	1*	0*	0*	0*	0*	1*		
							L1	L2	L3	L4	L5	L6
							1	0	0	0		
B^1	0*	0*	0*	0*	1*	1*	0*	0*	0*	1*		
C^1	0*	0*	0*	0*	1*	1*	0*	0*	0*	1*		
C^1	1*	0*	0*	0*	1*	0*	0*	0*	0*	1*		
D^1	0*	0*	0*	0*	1*	1*	0*	0*	0*	1*		
D^1	1*	0*	0*	0*	1*	0*	0*	0*	0*	1*		

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							0	1	0	1		
1												
D	0*	0*	0*	1*	1*	0*	1*	0*	0*	1*		
							L1	L2	L3	L4	L5	L6
							0	1	1	0		
1												
D	0*	0*	1*	0*	1*	0*	1*	0*	0*	1*		
							L1	L2	L3	L4	L5	L6
							0	2	0	0		
1												
C	0*	1*	0*	0*	1*	0*	1*	0*	0*	1*		
1												
D	0*	1*	0*	0*	1*	0*	1*	0*	0*	1*		
							L1	L2	L3	L4	L5	L6
							1	0	0	1		
1												
D	0*	0*	0*	1*	1*	1*	0*	0*	0*	1*		
							L1	L2	L3	L4	L5	L6
							1	0	1	0		
1												
D	0*	0*	1*	0*	1*	1*	0*	0*	0*	1*		
							L1	L2	L3	L4	L5	L6
							1	1	0	0		
1												
C	0*	1*	0*	0*	1*	1*	0*	0*	0*	1*		
1												
C	1*	0*	0*	0*	1*	0*	1*	0*	0*	1*		
1												
D	0*	1*	0*	0*	1*	1*	0*	0*	0*	1*		
1												
D	1*	0*	0*	0*	1*	0*	1*	0*	0*	1*		

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							2	0	0	0		
1												
C	1*	0*	0*	0*	1*	1*	0*	0*	0*	1*		
1												
D	1*	0*	0*	0*	1*	1*	0*	0*	0*	1*		
							L1	L2	L3	L4	L5	L6
							0	1	1	1		
1												
B	0*	0*	0*	0*	1*	0*	1*	1*	1*	1*		
1												
C	0*	0*	0*	0*	1*	0*	1*	1*	1*	1*		
1												
D	0*	0*	0*	0*	1*	0*	1*	1*	1*	1*		
							L1	L2	L3	L4	L5	L6
							0	1	2	0		
1												
B	0*	0*	0*	0*	1*	0*	1*	2*	0*	1*		
1												
C	0*	0*	0*	0*	1*	0*	1*	2*	0*	1*		
1												
D	0*	0*	0*	0*	1*	0*	1*	2*	0*	1*		
							L1	L2	L3	L4	L5	L6
							0	2	0	1		
1												
D	0*	2*	0*	1*	1*	0*	0*	0*	0*	1*		
							L1	L2	L3	L4	L5	L6
							0	2	1	0		
1												
B	0*	0*	0*	0*	1*	0*	2*	1*	0*	1*		
1												
C	0*	0*	0*	0*	1*	0*	2*	1*	0*	1*		
1												
D	0*	0*	0*	0*	1*	0*	2*	1*	0*	1*		

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

								L1	L2	L3	L4	L5	L6
								0	3	0	0		
	1												
B		0*	0*	0*	0*	1*	0*	3*	0*	0*	1*		
	1												
C		0*	0*	0*	0*	1*	0*	3*	0*	0*	1*		
	1												
D		0*	0*	0*	0*	1*	0*	3*	0*	0*	1*		
								L1	L2	L3	L4	L5	L6
								1	0	0	2		
	1												
B		0*	0*	0*	0*	1*	1*	0*	0*	2*	1*		
	1												
C		0*	0*	0*	0*	1*	1*	0*	0*	2*	1*		
	1												
D		0*	0*	0*	0*	1*	1*	0*	0*	2*	1*		
								L1	L2	L3	L4	L5	L6
								1	0	1	1		
	1												
B		0*	0*	0*	0*	1*	1*	0*	1*	1*	1*		
	1												
C		0*	0*	0*	0*	1*	1*	0*	1*	1*	1*		
	1												
D		0*	0*	0*	0*	1*	1*	0*	1*	1*	1*		
								L1	L2	L3	L4	L5	L6
								1	1	0	1		
	1												
B		0*	0*	0*	0*	1*	1*	1*	0*	1*	1*		
	1												
C		0*	0*	0*	0*	1*	1*	1*	0*	1*	1*		
	1												
D		0*	0*	0*	0*	1*	1*	1*	0*	1*	1*		
	1												
D		1*	1*	0*	1*	1*	0*	0*	0*	0*	1*		

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							1	1	1	0		
B^1	0*	0*	0*	0*	1*	1*	1*	1*	0*	1*		
C^1	0*	0*	0*	0*	1*	1*	1*	1*	0*	1*		
D^1	0*	0*	0*	0*	1*	1*	1*	1*	0*	1*		
D^1	1*	1*	1*	0*	1*	0*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							2	0	0	1		
B^1	0*	0*	0*	0*	1*	2*	0*	0*	1*	1*		
C^1	0*	0*	0*	0*	1*	2*	0*	0*	1*	1*		
D^1	0*	0*	0*	0*	1*	2*	0*	0*	1*	1*		

							L1	L2	L3	L4	L5	L6
							2	0	1	0		
D^1	2*	0*	1*	0*	1*	0*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							3	0	0	0		
B^1	0*	0*	0*	0*	1*	3*	0*	0*	0*	1*		
C^1	0*	0*	0*	0*	1*	3*	0*	0*	0*	1*		
D^1	0*	0*	0*	0*	1*	3*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							0	1	1	2		
D^1	0*	0*	0*	1*	1*	0*	1*	1*	1*	1*		

							L1	L2	L3	L4	L5	L6
							0	1	2	1		
D^1	0*	0*	0*	1*	1*	0*	1*	2*	0*	1*		
D^1	0*	0*	1*	0*	1*	0*	1*	1*	1*	1*		

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

```
                                                L1   L2   L3   L4   L5   L6
                                                 0    1    3    0
   1
D     0*   0*   1*   0*   1*   0*   1*   2*   0*   1*

                                                L1   L2   L3   L4   L5   L6
                                                 0    2    1    1
   1
C     0*   1*   0*   0*   1*   0*   1*   1*   1*   1*
   1
D     0*   0*   0*   1*   1*   0*   2*   1*   0*   1*
   1
D     0*   1*   0*   0*   1*   0*   1*   1*   1*   1*

                                                L1   L2   L3   L4   L5   L6
                                                 0    2    2    0
   1
C     0*   1*   0*   0*   1*   0*   1*   2*   0*   1*
   1
D     0*   0*   1*   0*   1*   0*   2*   1*   0*   1*
   1
D     0*   1*   0*   0*   1*   0*   1*   2*   0*   1*

                                                L1   L2   L3   L4   L5   L6
                                                 0    3    0    1
   1
D     0*   0*   0*   1*   1*   0*   3*   0*   0*   1*
   1
D     0*   2*   0*   1*   1*   0*   1*   0*   0*   1*

                                                L1   L2   L3   L4   L5   L6
                                                 0    3    1    0
   1
C     0*   1*   0*   0*   1*   0*   2*   1*   0*   1*
   1
D     0*   0*   1*   0*   1*   0*   3*   0*   0*   1*
   1
D     0*   1*   0*   0*   1*   0*   2*   1*   0*   1*
```

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							0	4	0	0		
C^1	0*	1*	0*	0*	1*	0*	3*	0*	0*	1*		
D^1	0*	1*	0*	0*	1*	0*	3*	0*	0*	1*		
							L1	L2	L3	L4	L5	L6
							1	0	0	3		
D^1	0*	0*	0*	1*	1*	1*	0*	0*	2*	1*		
							L1	L2	L3	L4	L5	L6
							1	0	1	2		
D^1	0*	0*	0*	1*	1*	1*	0*	1*	1*	1*		
D^1	0*	0*	1*	0*	1*	1*	0*	0*	2*	1*		
							L1	L2	L3	L4	L5	L6
							1	0	2	1		
D^1	0*	0*	1*	0*	1*	1*	0*	1*	1*	1*		
							L1	L2	L3	L4	L5	L6
							1	1	0	2		
C^1	0*	1*	0*	0*	1*	1*	0*	0*	2*	1*		
D^1	0*	0*	0*	1*	1*	1*	1*	0*	1*	1*		
D^1	0*	1*	0*	0*	1*	1*	0*	0*	2*	1*		
							L1	L2	L3	L4	L5	L6
							1	1	1	1		
C^1	0*	1*	0*	0*	1*	1*	0*	1*	1*	1*		
C^1	1*	0*	0*	0*	1*	0*	1*	1*	1*	1*		
D^1	0*	0*	0*	1*	1*	1*	1*	1*	0*	1*		
D^1	0*	0*	1*	0*	1*	1*	1*	0*	1*	1*		
D^1	0*	1*	0*	0*	1*	1*	0*	1*	1*	1*		
D^1	1*	0*	0*	0*	1*	0*	1*	1*	1*	1*		

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

```
                                                  L1   L2   L3   L4   L5   L6
                                                   1    1    2    0
      1
C       1*   0*   0*   0*   1*   0*   1*   2*   0*   1*
      1
D       0*   0*   1*   0*   1*   1*   1*   1*   0*   1*
      1
D       1*   0*   0*   0*   1*   0*   1*   2*   0*   1*

                                                  L1   L2   L3   L4   L5   L6
                                                   1    2    0    1
      1
C       0*   1*   0*   0*   1*   1*   1*   0*   1*   1*
      1
D       0*   1*   0*   0*   1*   1*   1*   0*   1*   1*
      1
D       0*   2*   0*   1*   1*   1*   0*   0*   0*   1*
      1
D       1*   1*   0*   1*   1*   0*   1*   0*   0*   1*

                                                  L1   L2   L3   L4   L5   L6
                                                   1    2    1    0
      1
C       0*   1*   0*   0*   1*   1*   1*   1*   0*   1*
      1
C       1*   0*   0*   0*   1*   0*   2*   1*   0*   1*
      1
D       0*   1*   0*   0*   1*   1*   1*   1*   0*   1*
      1
C       1*   0*   0*   0*   1*   0*   2*   1*   0*   1*
      1
D       1*   1*   1*   0*   1*   0*   1*   0*   0*   1*

                                                  L1   L2   L3   L4   L5   L6
                                                   1    3    0    0
      1
C       1*   0*   0*   0*   1*   0*   3*   0*   0*   1*
      1
C       1*   0*   0*   0*   1*   0*   3*   0*   0*   1*
```

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

								L1	L2	L3	L4	L5	L6
								2	0	C	2		
C	1	1*	0*	0*	0*	1*	1*	0*	0*	2*	1*		
D	1	0*	0*	0*	1*	1*	2*	0*	0*	1*	1*		
D	1	1*	0*	0*	0*	1*	1*	0*	0*	2*	1*		
								L1	L2	L3	L4	L5	L6
								2	0	1	1		
C	1	1*	0*	0*	0*	1*	1*	0*	1*	1*	1*		
D	1	0*	0*	1*	0*	1*	2*	0*	0*	1*	1*		
D	1	1*	0*	0*	0*	1*	1*	0*	1*	1*	1*		
								L1	L2	L3	L4	L5	L6
								2	1	0	1		
C	1	0*	1*	0*	0*	1*	2*	0*	0*	1*	1*		
C	1	1*	0*	0*	0*	1*	1*	1*	0*	1*	1*		
D	1	0*	1*	0*	0*	1*	2*	0*	0*	1*	1*		
D	1	1*	0*	0*	0*	1*	1*	1*	0*	1*	1*		
D	1	1*	1*	0*	1*	1*	1*	0*	0*	0*	1*		
								L1	L2	L3	L4	L5	L6
								2	1	1	0		
C	1	1*	0*	0*	0*	1*	1*	1*	1*	0*	1*		
D	1	1*	0*	0*	0*	1*	1*	1*	1*	0*	1*		
D	1	1*	1*	1*	0*	1*	1*	0*	0*	0*	1*		
D	1	2*	0*	1*	0*	1*	0*	1*	0*	0*	1*		

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							3	0	0	1		
C^1	1*	0*	0*	0*	1*	2*	0*	0*	1*	1*		
C^1	0*	0*	0*	1*	1*	3*	0*	0*	0*	1*		
D^1	1*	0*	0*	0*	1*	2*	0*	0*	1*	1*		

							L1	L2	L3	L4	L5	L6
							3	0	1	0		
D^1	0*	0*	1*	0*	1*	3*	0*	0*	0*	1*		
D^1	2*	0*	1*	0*	1*	1*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							3	1	0	0		
C^1	0*	1*	0*	0*	1*	3*	0*	0*	0*	1*		
D^1	0*	1*	0*	0*	1*	3*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							4	0	0	0		
C^1	1*	0*	0*	0*	1*	3*	0*	0*	0*	1*		
D^1	1*	0*	0*	0*	1*	3*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							1	2	0	2		
B^1	0*	0*	0*	0*	1*	1*	2*	0*	2*	1*		
C^1	0*	0*	0*	0*	1*	1*	2*	0*	2*	1*		
D^1	0*	0*	0*	0*	1*	1*	2*	0*	2*	1*		

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

						L1	L2	L3	L4	L5	L6
						1	2	1	1		
B [1]	0*	0*	0*	0*	1*	1*	2*	1*	1*	1*	
C [1]	0*	0*	0*	0*	1*	1*	2*	1*	1*	1*	
D [1]	0*	0*	0*	0*	1*	1*	2*	1*	1*	1*	

						L1	L2	L3	L4	L5	L6
						1	2	2	0		
B [1]	0*	0*	0*	0*	1*	1*	2*	2*	0*	1*	
C [1]	0*	0*	0*	0*	1*	1*	2*	2*	0*	1*	
D [1]	0*	0*	0*	0*	1*	1*	2*	2*	0*	1*	

						L1	L2	L3	L4	L5	L6
						2	1	0	2		
B [1]	0*	0*	0*	0*	1*	2*	1*	0*	2*	1*	
C [1]	0*	0*	0*	0*	1*	2*	1*	0*	2*	1*	
D [1]	0*	0*	0*	0*	1*	2*	1*	0*	2*	1*	

						L1	L2	L3	L4	L5	L6
						2	1	1	1		
B [1]	0*	0*	0*	0*	1*	2*	1*	1*	1*	1*	
C [1]	0*	0*	0*	0*	1*	2*	1*	1*	1*	1*	
D [1]	0*	0*	0*	0*	1*	2*	1*	1*	1*	1*	

						L1	L2	L3	L4	L5	L6
						2	1	2	0		
B [1]	0*	0*	0*	0*	1*	2*	1*	2*	0*	1*	
C [1]	0*	0*	0*	0*	1*	2*	1*	2*	0*	1*	
D [1]	0*	0*	0*	0*	1*	2*	1*	2*	0*	1*	

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

```
                                           L1   L2   L3   L4   L5   L6
                                            0    3    1    2
     1
D      0*   2*   0*   1*   1*   0*   1*   1*   1*   1*

                                           L1   L2   L3   L4   L5   L6
                                            0    3    2    1
     1
D      0*   2*   0*   1*   1*   0*   1*   2*   0*   1*

                                           L1   L2   L3   L4   L5   L6
                                            0    4    1    1
     1
D      0*   2*   0*   1*   1*   0*   2*   1*   0*   1*

                                           L1   L2   L3   L4   L5   L6
                                            0    5    0    1
     1
D      0*   2*   0*   1*   1*   0*   3*   0*   0*   1*

                                           L1   L2   L3   L4   L5   L6
                                            1    2    0    3
     1
D      0*   0*   0*   1*   1*   1*   2*   0*   2*   1*
     1
D      0*   2*   0*   1*   1*   1*   0*   0*   2*   1*

                                           L1   L2   L3   L4   L5   L6
                                            1    2    1    2
     1
D      0*   0*   0*   1*   1*   1*   2*   1*   1*   1*
     1
D      0*   0*   1*   0*   1*   1*   2*   0*   2*   1*
     1
D      0*   2*   0*   1*   1*   1*   0*   1*   1*   1*
     1
D      1*   1*   0*   1*   1*   0*   1*   1*   1*   1*
```

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							1	2	2	1		
1												
C	0*	0*	0*	1*	1*	1*	2*	2*	0*	1*		
1												
D	0*	0*	1*	0*	1*	1*	2*	1*	1*	1*		
1												
C	1*	1*	0*	1*	1*	0*	1*	2*	0*	1*		
1												
C	1*	1*	1*	0*	1*	0*	1*	1*	1*	1*		

							L1	L2	L3	L4	L5	L6
							1	2	3	0		
1												
D	0*	0*	1*	0*	1*	1*	2*	2*	0*	1*		
1												
D	1*	1*	1*	0*	1*	0*	1*	2*	0*	1*		

							L1	L2	L3	L4	L5	L6
							1	3	0	2		
1												
C	0*	1*	0*	0*	1*	1*	2*	0*	2*	1*		
1												
C	0*	1*	0*	0*	1*	1*	2*	0*	2*	1*		
1												
C	0*	2*	0*	1*	1*	1*	1*	0*	1*	1*		

							L1	L2	L3	L4	L5	L6
							1	3	1	1		
1												
C	0*	1*	0*	0*	1*	1*	2*	1*	1*	1*		
1												
D	0*	1*	0*	0*	1*	1*	2*	1*	1*	1*		
1												
D	0*	2*	0*	1*	1*	1*	1*	1*	0*	1*		
1												
D	1*	1*	0*	1*	1*	0*	2*	1*	0*	1*		

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

```
                                         L1  L2  L3  L4  L5  L6
                                          1   3   2   0
    1
C      0*  1*  0*  0*  1*  1*  2*  2*  0*  1*
    1
D      0*  1*  0*  0*  1*  1*  2*  2*  0*  1*
    1
D      1*  1*  1*  0*  1*  0*  2*  1*  0*  1*
                                         L1  L2  L3  L4  L5  L6
                                          1   4   0   1
    1
D      1*  1*  0*  1*  1*  0*  3*  0*  0*  1*
                                         L1  L2  L3  L4  L5  L6
                                          1   4   1   0
    1
D      1*  1*  1*  0*  1*  0*  3*  0*  0*  1*
                                         L1  L2  L3  L4  L5  L6
                                          2   1   0   3
    1
D      0*  0*  0*  1*  1*  2*  1*  0*  2*  1*
    1
D      1*  1*  0*  1*  1*  1*  0*  0*  2*  1*
                                         L1  L2  L3  L4  L5  L6
                                          2   1   1   2
    1
D      0*  0*  0*  1*  1*  2*  1*  1*  1*  1*
    1
D      0*  0*  1*  0*  1*  2*  1*  0*  2*  1*
    1
D      1*  1*  0*  1*  1*  1*  0*  1*  1*  1*
    1
D      1*  1*  1*  0*  1*  1*  0*  0*  2*  1*
```

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

L1	L2	L3	L4	L5	L6
2	1	2	1		

D^{1}	0*	0*	0*	1*	1*	2*	1*	2*	0*	1*
D^{1}	0*	0*	1*	0*	1*	2*	1*	1*	1*	1*
C^{1}	1*	1*	1*	0*	1*	1*	0*	1*	1*	1*
D^{1}	2*	0*	1*	0*	1*	0*	1*	1*	1*	1*

L1	L2	L3	L4	L5	L6
2	1	3	0		

D^{1}	0*	0*	1*	0*	1*	2*	1*	2*	0*	1*
D^{1}	2*	0*	1*	0*	1*	0*	1*	2*	0*	1*

L1	L2	L3	L4	L5	L6
2	2	0	2		

C^{1}	0*	1*	0*	0*	1*	2*	1*	0*	2*	1*
C^{1}	1*	0*	0*	0*	1*	1*	2*	0*	2*	1*
D^{1}	0*	1*	0*	0*	1*	2*	1*	0*	2*	1*
D^{1}	0*	2*	0*	1*	1*	2*	0*	0*	1*	1*
C^{1}	1*	0*	0*	0*	1*	1*	2*	0*	2*	1*
D^{1}	1*	1*	0*	1*	1*	1*	1*	0*	1*	1*

L1	L2	L3	L4	L5	L6
2	2	1	1		

C^{1}	0*	1*	0*	0*	1*	2*	1*	1*	1*	1*
C^{1}	1*	0*	0*	0*	1*	1*	2*	1*	1*	1*
D^{1}	0*	1*	0*	0*	1*	2*	1*	1*	1*	1*
C^{1}	1*	0*	0*	0*	1*	1*	2*	1*	1*	1*
D^{1}	1*	1*	0*	1*	1*	1*	1*	1*	0*	1*
D^{1}	1*	1*	1*	0*	1*	1*	1*	0*	1*	1*

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

```
                                        L1   L2   L3   L4   L5   L6
                                         2    2    2    0
     1
C      0*   1*   0*   0*   1*   2*   1*   2*   0*   1*
     1
C      1*   0*   0*   0*   1*   1*   2*   2*   0*   1*
     1
D      0*   1*   0*   0*   1*   2*   1*   2*   0*   1*
     1
D      1*   0*   0*   0*   1*   1*   2*   2*   0*   1*
     1
D      1*   1*   1*   0*   1*   1*   1*   1*   0*   1*
     1
D      2*   0*   1*   0*   1*   0*   2*   1*   0*   1*

                                        L1   L2   L3   L4   L5   L6
                                         2    3    1    0
     1
D      2*   0*   1*   0*   1*   0*   3*   0*   0*   1*

                                        L1   L2   L3   L4   L5   L6
                                         3    0    1    2
     1
D      2*   0*   1*   0*   1*   1*   0*   0*   2*   1*

                                        L1   L2   L3   L4   L5   L6
                                         2    0    2    1
     1
D      2*   0*   1*   0*   1*   1*   0*   1*   1*   1*

                                        L1   L2   L3   L4   L5   L6
                                         3    1    0    2
     1
C      1*   0*   0*   0*   1*   2*   1*   0*   2*   1*
     1
D      1*   0*   0*   0*   1*   2*   1*   0*   2*   1*
     1
D      1*   1*   0*   1*   1*   2*   0*   0*   1*   1*
```

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

								L1	L2	L3	L4	L5	L6
								3	1	1	1		
	1												
C		1*	0*	0*	0*	1*	2*	1*	1*	1*	1*		
	1												
D		1*	0*	0*	0*	1*	2*	1*	1*	1*	1*		
	1												
D		1*	1*	1*	0*	1*	2*	0*	0*	1*	1*		
	1												
D		2*	0*	1*	0*	1*	1*	1*	0*	1*	1*		
								L1	L2	L3	L4	L5	L6
								3	1	2	0		
	1												
C		1*	0*	0*	0*	1*	2*	1*	2*	0*	1*		
	1												
D		1*	0*	0*	0*	1*	2*	1*	2*	0*	1*		
	1												
D		2*	0*	1*	0*	1*	1*	1*	1*	0*	1*		
								L1	L2	L3	L4	L5	L6
								3	2	0	1		
	1												
D		0*	2*	0*	1*	1*	3*	0*	0*	0*	1*		
								L1	L2	L3	L4	L5	L6
								4	0	1	1		
	1												
C		2*	0*	1*	0*	1*	2*	0*	0*	1*	1*		
								L1	L2	L3	L4	L5	L6
								4	1	0	1		
	1												
D		1*	1*	0*	1*	1*	3*	0*	0*	0*	1*		
								L1	L2	L3	L4	L5	L6
								4	1	1	0		
	1												
D		1*	1*	1*	0*	1*	3*	0*	0*	0*	1*		

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

```
                                          U1  L2  L3  L4  L5  L6
   1                                       5   0   1   0
C    2*  0*  1*  0*  1*  3*  0*  0*  0*  1*

                                          U1  L2  L3  L4  L5  L6
   1                                       1   4   0   3
C    0*  2*  0*  1*  1*  1*  2*  0*  2*  1*

                                          U1  L2  L3  L4  L5  L6
   1                                       1   4   1   2
D    0*  2*  0*  1*  1*  1*  2*  1*  1*  1*

                                          U1  L2  L3  L4  L5  L6
   1                                       1   4   2   1
D    0*  2*  0*  1*  1*  1*  2*  2*  0*  1*

                                          L1  L2  L3  L4  L5  L6
   1                                       2   3   0   ?
C    0*  2*  0*  1*  1*  2*  1*  0*  2*  1*
   1
D    1*  1*  0*  1*  1*  1*  2*  0*  2*  1*

                                          U1  L2  L3  L4  L5  L6
   1                                       2   3   1   2
D    0*  2*  0*  1*  1*  2*  1*  1*  1*  1*
   1
C    1*  1*  0*  1*  1*  1*  2*  1*  1*  1*
   1
D    1*  1*  1*  0*  1*  1*  2*  0*  2*  1*
```

Table 2-2. SERIES FOR RADIUS CONDITION (continued)

```
                                          L1  L2  L3  L4  L5  L6
                                           2   3   2   1
   1
D    0*  2*  0*  1*  1*  2*  1*  2*  0*  1*
   1
D    1*  1*  0*  1*  1*  1*  2*  2*  0*  1*
   1
D    1*  1*  1*  0*  1*  1*  2*  1*  1*  1*
                                          L1  L2  L3  L4  L5  L6
                                           2   3   3   0
   1
D    1*  1*  1*  0*  1*  1*  2*  2*  0*  1*
                                          L1  L2  L3  L4  L5  L6
                                           3   2   0   3
   1
D    1*  1*  0*  1*  1*  2*  1*  0*  2*  1*
                                          L1  L2  L3  L4  L5  L6
                                           3   2   1   2
   1
D    1*  1*  0*  1*  1*  2*  1*  1*  1*  1*
   1
D    1*  1*  1*  0*  1*  2*  1*  0*  2*  1*
   1
D    2*  0*  1*  0*  1*  1*  2*  0*  2*  1*
                                          L1  L2  L3  L4  L5  L6
                                           3   2   2   1
   1
D    1*  1*  0*  1*  1*  2*  1*  2*  0*  1*
   1
D    1*  1*  1*  0*  1*  2*  1*  1*  1*  1*
   1
D    2*  0*  1*  0*  1*  1*  2*  1*  1*  1*
                                          L1  L2  L3  L4  L5  L6
                                           3   2   3   0
   1
D    1*  1*  1*  0*  1*  2*  1*  2*  0*  1*
   1
D    2*  0*  1*  0*  1*  1*  2*  2*  0*  1*
```

Table 2-2. SERIES FOR RADIUS CONDITION (concluded)

```
                                          L1   L2   L3   L4   L5   L6
                                           4    1    1    2
   1
D     2*   0*   1*   0*   1*   2*   1*   0*   2*   1*

                                          L1   L2   L3   L4   L5   L6
                                           4    1    2    1
   1
D     2*   0*   1*   0*   1*   2*   1*   1*   1*   1*

                                          L1   L2   L3   L4   L5   L6
                                           4    1    3    0
   1
D     2*   0*   1*   0*   1*   2*   1*   2*   0*   1*

************** ***********************************************

SAVE TAPE
CASE ID=  X**2+Y**2-RCO**2
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION

```
                                         L1   L2   L3   L4   L5   L6
                                          0    0    0    0
         1
   A        0*   0*   0*   0*   1*
         1
   B        0*   0*   0*   0*   1*   0*   0*   0*   0*   1*
         1
   C        0*   0*   0*   0*   1*   0*   0*   0*   0*   1*
         1
   D        0*   0*   0*   0*   1*   0*   0*   0*   0*   1*

                                         L1   L2   L3   L4   L5   L6
                                          0    1    0    0
         1
   B        0*   0*   0*   0*   1*   0*   1*   0*   0*   1*
         1
   B        0*   1*   0*   0*   1*   0*   0*   0*   0*   1*
         1
   C        0*   0*   0*   0*   1*   0*   1*   0*   0*   1*
         1
   C        0*   1*   0*   0*   1*   0*   0*   0*   0*   1*
         1
   D        0*   0*   0*   0*   1*   0*   1*   0*   0*   1*
         1
   D        0*   1*   0*   0*   1*   0*   0*   0*   0*   1*

                                         L1   L2   L3   L4   L5   L6
                                          1    0    0    0
         1
   B        0*   0*   0*   0*   1*   1*   0*   0*   0*   1*
         1
   B        1*   0*   0*   0*   1*   0*   0*   0*   0*   1*
         1
   C        0*   0*   0*   0*   1*   1*   0*   0*   0*   1*
         1
   C        1*   0*   0*   0*   1*   0*   0*   0*   0*   1*
         1
   D        0*   0*   0*   0*   1*   1*   0*   0*   0*   1*
         1
   D        1*   0*   0*   0*   1*   0*   0*   0*   0*   1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							0	2	0	0		
B_1	0*	1*	0*	0*	1*	0*	1*	0*	0*	1*		
C_1	0*	1*	0*	0*	1*	0*	1*	0*	0*	1*		
D_1	0*	2*	0*	0*	1*	0*	1*	0*	0*	1*		
D_1	0*	2*	0*	0*	1*	0*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							1	1	0	0		
B_1	0*	1*	0*	0*	1*	1*	0*	0*	0*	1*		
B_1	1*	0*	0*	0*	1*	0*	1*	0*	0*	1*		
C_1	0*	1*	0*	0*	1*	1*	0*	0*	0*	1*		
C_1	1*	0*	0*	0*	1*	0*	1*	0*	0*	1*		
D_1	0*	1*	0*	0*	1*	1*	0*	0*	0*	1*		
D_1	1*	0*	0*	0*	1*	0*	1*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							2	0	0	0		
B_1	1*	0*	0*	0*	1*	1*	0*	0*	0*	1*		
C_1	1*	0*	0*	0*	1*	1*	0*	0*	0*	1*		
D_1	1*	0*	0*	0*	1*	1*	0*	0*	0*	1*		
D_1	2*	0*	0*	0*	1*	0*	0*	0*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							0	1	1	1		
B^1	0*	0*	0*	0*	1*	0*	1*	1*	1*	1*		
C^1	0*	0*	0*	0*	1*	0*	1*	1*	1*	1*		
D^1	0*	0*	0*	0*	1*	0*	1*	1*	1*	1*		
D^1	0*	1*	1*	1*	1*	0*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							0	1	2	0		
B^1	0*	0*	0*	0*	1*	0*	1*	2*	0*	1*		
C^1	0*	0*	0*	0*	1*	0*	1*	2*	0*	1*		
D^1	0*	0*	0*	0*	1*	0*	1*	2*	0*	1*		
D^1	0*	1*	2*	0*	1*	0*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							[illegible]	2	1	0		
B^1	0*	0*	0*	0*	1*	0*	2*	1*	0*	1*		
C^1	0*	0*	0*	0*	1*	0*	2*	1*	0*	1*		
C^1	0*	2*	1*	0*	1*	0*	0*	0*	0*	1*		
D^1	0*	0*	0*	0*	1*	0*	2*	1*	0*	1*		
D^2	0*	2*	1*	0*	1*	0*	0*	0*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                         L1   L2   L3   L4   L5   L6
                                         0    2    0    0
     1
B       0*   0*   0*   0*   1*   0*   3*   0*   0*   1*
     1
C       0*   0*   0*   0*   1*   0*   3*   0*   0*   1*
     1
D       0*   0*   0*   0*   1*   0*   3*   0*   0*   1*
     1
D       0*   2*   0*   0*   1*   0*   1*   0*   0*   1*
     1
D       0*   3*   0*   0*   1*   0*   0*   0*   0*   1*

                                         L1   L2   L3   L4   L5   L6
                                         1    0    0    2
     1
B       0*   0*   0*   0*   1*   1*   0*   0*   2*   1*
     1
C       0*   0*   0*   0*   1*   1*   0*   0*   2*   1*
     1
D       0*   0*   0*   0*   1*   1*   0*   0*   2*   1*
     1
D       1*   0*   0*   2*   1*   0*   0*   0*   0*   1*

                                         L1   L2   L3   L4   L5   L6
                                         1    0    1    1
     1
B       0*   0*   0*   0*   1*   1*   0*   1*   1*   1*
     1
C       0*   0*   0*   0*   1*   1*   0*   1*   1*   1*
     1
D       0*   0*   0*   0*   1*   1*   0*   1*   1*   1*
     1
D       1*   0*   1*   1*   1*   0*   0*   0*   0*   1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							1	1	0	1		
B^1	0*	0*	0*	0*	1*	1*	1*	0*	1*	1*		
C^1	0*	0*	0*	0*	1*	1*	1*	0*	1*	1*		
C^1	1*	1*	0*	1*	1*	0*	0*	0*	0*	1*		
D^1	0*	0*	0*	0*	1*	1*	1*	0*	1*	1*		
D^1	1*	1*	0*	1*	1*	0*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							1	1	1	0		
B^1	0*	0*	0*	0*	1*	1*	1*	1*	0*	1*		
C^1	0*	0*	0*	0*	1*	1*	1*	1*	0*	1*		
C^1	1*	1*	1*	0*	1*	0*	0*	0*	0*	1*		
D^1	0*	0*	0*	0*	1*	1*	1*	1*	0*	1*		
D^1	1*	1*	1*	0*	1*	0*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							1	2	0	0		
D^1	0*	2*	0*	0*	1*	1*	0*	0*	0*	1*		
D^1	1*	2*	0*	0*	1*	0*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							2	0	0	1		
B^1	0*	0*	0*	0*	1*	2*	0*	0*	1*	1*		
C^1	0*	0*	0*	0*	1*	2*	0*	0*	1*	1*		
C^1	2*	0*	0*	1*	1*	0*	0*	0*	0*	1*		
D^1	0*	0*	0*	0*	1*	2*	0*	0*	1*	1*		
D^1	2*	0*	0*	1*	1*	0*	0*	0*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							2	1	0	0		
D^1	2*	0*	0*	0*	1*	0*	1*	0*	0*	1*		
D^2	2*	1*	0*	0*	1*	0*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							3	0	0	0		
B^1	0*	0*	0*	0*	1*	3*	0*	0*	0*	1*		
C^1	0*	0*	0*	0*	1*	3*	0*	0*	0*	1*		
D^1	0*	0*	0*	0*	1*	3*	0*	0*	0*	1*		
D^1	2*	0*	0*	0*	1*	1*	0*	0*	0*	1*		
D^1	3*	0*	0*	0*	1*	0*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							0	2	1	1		
B^1	0*	1*	0*	0*	1*	0*	1*	1*	1*	1*		
C^1	0*	1*	0*	0*	1*	0*	1*	1*	1*	1*		
D^1	0*	1*	0*	0*	1*	0*	1*	1*	1*	1*		
D^1	0*	1*	1*	1*	1*	0*	1*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							0	2	2	0		
B^1	0*	1*	0*	0*	1*	0*	1*	2*	0*	1*		
C^1	0*	1*	0*	0*	1*	0*	1*	2*	0*	1*		
D^1	0*	1*	0*	0*	1*	0*	1*	2*	0*	1*		
C^1	0*	1*	2*	0*	1*	0*	1*	0*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

L1	L2	L3	L4	L5	L6
0	3	1	0		

B	1	0*	1*	0*	0*	1*	0*	2*	1*	0*	1*
C	1	0*	1*	0*	0*	1*	0*	2*	1*	0*	1*
C	1	0*	2*	1*	0*	1*	0*	1*	0*	0*	1*
D	1	0*	1*	0*	0*	1*	0*	2*	1*	0*	1*
D	2	0*	2*	1*	0*	1*	0*	1*	0*	0*	1*

L1	L2	L3	L4	L5	L6
0	4	0	0		

B	1	0*	1*	0*	0*	1*	0*	3*	0*	0*	1*
C	1	0*	1*	0*	0*	1*	0*	3*	0*	0*	1*
D	1	0*	1*	0*	0*	1*	0*	3*	0*	0*	1*
D	1	0*	3*	0*	0*	1*	0*	1*	0*	0*	1*

L1	L2	L3	L4	L5	L6
1	1	0	2		

B	1	0*	1*	0*	0*	1*	1*	0*	0*	2*	1*
C	1	0*	1*	0*	0*	1*	1*	0*	0*	2*	1*
D	1	0*	1*	0*	0*	1*	1*	0*	0*	2*	1*
D	1	1*	0*	0*	2*	1*	0*	1*	0*	0*	1*

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							1	1	1	1		
B^{1}	0*	1*	0*	0*	1*	1*	0*	1*	1*	1*		
~~B^{1}~~	~~1*~~	~~0*~~	~~0*~~	~~0*~~	~~1*~~	~~0*~~	~~1*~~	~~1*~~	~~1*~~	~~1*~~		
C^{1}	0*	1*	0*	0*	1*	1*	0*	1*	1*	1*		
C^{1}	1*	0*	0*	0*	1*	0*	1*	1*	1*	1*		
~~D^{1}~~	~~0*~~	~~1*~~	~~0*~~	~~0*~~	~~1*~~	~~1*~~	~~0*~~	~~1*~~	~~1*~~	~~1*~~		
D^{1}	0*	1*	1*	1*	1*	1*	0*	0*	0*	1*		
D^{1}	1*	0*	0*	0*	1*	0*	1*	1*	1*	1*		
~~D^{1}~~	~~1*~~	~~0*~~	~~1*~~	~~1*~~	~~1*~~	~~0*~~	~~1*~~	~~0*~~	~~0*~~	~~1*~~		

							L1	L2	L3	L4	L5	L6
							1	1	2	0		
B^{1}	1*	0*	0*	0*	1*	0*	1*	2*	0*	1*		
C^{1}	1*	0*	0*	0*	1*	0*	1*	2*	0*	1*		
D^{1}	0*	1*	2*	0*	1*	1*	0*	0*	0*	1*		
D^{1}	1*	0*	0*	0*	1*	0*	1*	2*	0*	1*		

							L1	L2	L3	L4	L5	L6
							1	2	0	1		
B^{1}	0*	1*	0*	0*	1*	1*	1*	0*	1*	1*		
~~C^{1}~~	~~0*~~	~~1*~~	~~0*~~	~~0*~~	~~1*~~	~~1*~~	~~1*~~	~~0*~~	~~1*~~	~~1*~~		
C^{1}	1*	1*	0*	1*	1*	0*	1*	0*	0*	1*		
~~C^{1}~~	~~0*~~	~~1*~~	~~0*~~	~~0*~~	1*	1*	1*	0*	1*	1*		
~~D^{1}~~	~~1*~~	~~1*~~	~~0*~~	~~1*~~	~~1*~~	~~0*~~	1*	0*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							1	2	1	0		
B_1	0*	1*	0*	0*	1*	1*	1*	1*	0*	1*		
B_1	1*	0*	0*	0*	1*	0*	2*	1*	0*	1*		
C_1	0*	1*	0*	0*	1*	1*	1*	1*	0*	1*		
C_1	0*	2*	1*	0*	1*	1*	0*	0*	0*	1*		
C_1	1*	0*	0*	0*	1*	0*	2*	1*	0*	1*		
C_1	1*	1*	1*	0*	1*	0*	1*	0*	0*	1*		
D_1	0*	1*	0*	0*	1*	1*	1*	1*	0*	1*		
D_2	0*	2*	1*	0*	1*	1*	0*	0*	0*	1*		
D_1	1*	0*	0*	0*	1*	0*	2*	1*	0*	1*		
D_1	1*	1*	1*	0*	1*	0*	1*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							1	3	0	0		
B_1	1*	0*	0*	0*	1*	0*	2*	0*	0*	1*		
C_1	1*	0*	0*	0*	1*	0*	2*	0*	0*	1*		
C_1	0*	3*	0*	0*	1*	1*	0*	0*	0*	1*		
D_1	1*	0*	0*	0*	1*	0*	3*	0*	0*	1*		
D_1	1*	2*	0*	0*	1*	0*	1*	0*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							2	0	0	2		
B^{1}	1*	0*	0*	0*	1*	1*	0*	0*	2*	1*		
C^{1}	1*	0*	0*	0*	1*	1*	0*	0*	2*	1*		
D^{1}	1*	0*	0*	0*	1*	1*	0*	0*	2*	1*		
D^{1}	1*	0*	0*	2*	1*	1*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							2	0	1	1		
B^{1}	1*	0*	0*	0*	1*	1*	0*	1*	1*	1*		
C^{1}	1*	0*	0*	0*	1*	1*	0*	1*	1*	1*		
D^{1}	1*	0*	0*	0*	1*	1*	0*	1*	1*	1*		
D^{1}	1*	0*	1*	1*	1*	1*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							2	1	0	1		
B^{1}	0*	1*	0*	0*	1*	2*	0*	0*	1*	1*		
B^{1}	1*	0*	0*	0*	1*	1*	1*	0*	1*	1*		
C^{1}	0*	1*	0*	0*	1*	2*	0*	0*	1*	1*		
C^{1}	1*	0*	0*	0*	1*	1*	1*	0*	1*	1*		
C^{1}	1*	1*	0*	1*	1*	1*	0*	0*	0*	1*		
C^{1}	2*	0*	0*	1*	1*	0*	1*	0*	0*	1*		
D^{1}	0*	1*	0*	0*	1*	2*	0*	0*	1*	1*		
D^{1}	1*	0*	0*	0*	1*	1*	1*	0*	1*	1*		
D^{1}	1*	1*	0*	1*	1*	1*	0*	0*	0*	1*		
D^{1}	2*	0*	0*	1*	1*	0*	1*	0*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                          L1   L2   L3   L4   L5   L6
                                           2    1    1    0
      1
   B     1*   0*   0*   0*   1*   1*   1*   1*   0*   1*
      1
   C     1*   0*   0*   0*   1*   1*   1*   1*   0*   1*
      1
   C     1*   1*   1*   0*   1*   1*   0*   0*   0*   1*
      1
   D     1*   0*   0*   0*   1*   1*   1*   1*   0*   1*
      1
   D     1*   1*   1*   0*   1*   1*   0*   0*   0*   1*

                                          L1   L2   L3   L4   L5   L6
                                           2    2    0    0
      1
   C     1*   2*   0*   0*   1*   1*   0*   0*   0*   1*
      2
   D     2*   1*   0*   0*   1*   0*   1*   0*   0*   1*

                                          L1   L2   L3   L4   L5   L6
                                           3    0    0    1
      1
   B     1*   0*   0*   0*   1*   2*   0*   0*   1*   1*
      1
   C     1*   0*   0*   0*   1*   2*   0*   0*   1*   1*
      1
   C     2*   0*   0*   1*   1*   1*   0*   0*   0*   1*
      1
   D     1*   0*   0*   0*   1*   2*   0*   0*   1*   1*
      1
   D     2*   0*   0*   1*   1*   1*   0*   0*   0*   1*

                                          L1   L2   L3   L4   L5   L6
                                           3    1    0    0
      1
   B     0*   1*   0*   0*   1*   3*   0*   0*   0*   1*
      1
   C     0*   1*   0*   0*   1*   3*   0*   0*   0*   1*
      1
   D     0*   1*   0*   0*   1*   3*   0*   0*   0*   1*
      2
   D     2*   1*   0*   0*   1*   1*   0*   0*   0*   1*
      1
   D     3*   0*   0*   0*   1*   0*   1*   0*   0*   1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

		L1	L2	L3	L4	L5	L6
		4	0	0	0		

B^1	1*	0*	0*	0*	1*	3*	0*	0*	0*	1*
C^1	1*	0*	0*	0*	1*	3*	0*	0*	0*	1*
D^1	1*	0*	0*	0*	1*	3*	0*	0*	0*	1*
D^1	3*	0*	0*	0*	1*	1*	0*	0*	0*	1*

		L1	L2	L3	L4	L5	L6
		0	3	1	1		

D^1	0*	2*	0*	0*	1*	0*	1*	1*	1*	1*

		L1	L2	L3	L4	L5	L6
		0	3	2	0		

D^1	0*	2*	0*	0*	1*	0*	1*	2*	0*	1*

		L1	L2	L3	L4	L5	L6
		0	4	1	0		

D^1	0*	2*	0*	0*	1*	0*	2*	1*	0*	1*

		L1	L2	L3	L4	L5	L6
		0	5	0	0		

D^1	0*	2*	0*	0*	1*	0*	3*	0*	0*	1*

		L1	L2	L3	L4	L5	L6
		1	2	0	2		

B^1	0*	0*	0*	0*	1*	1*	2*	0*	2*	1*
C^1	0*	0*	0*	0*	1*	1*	2*	0*	2*	1*
D^1	0*	0*	0*	0*	1*	1*	2*	0*	2*	1*
D^1	0*	2*	0*	0*	1*	1*	0*	0*	2*	1*
D^1	1*	2*	0*	2*	1*	0*	0*	0*	0*	1*

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							1	2	1	1		
B^1	C*	C*	0*	0*	1*	1*	2*	1*	1*	1*		
C^1	C*	C*	0*	0*	1*	1*	2*	1*	1*	1*		
D^1	C*	C*	0*	C*	1*	1*	2*	1*	1*	1*		
C^1	C*	2*	C*	0*	1*	1*	0*	1*	1*	1*		
D^1	1*	2*	1*	1*	1*	0*	0*	C*	C*	1*		

							L1	L2	L3	L4	L5	L6
							1	2	2	0		
B^1	C*	C*	C*	C*	1*	1*	2*	2*	C*	1*		
C^1	C*	C*	0*	0*	1*	1*	2*	2*	C*	1*		
D^1	C*	0*	C*	0*	1*	1*	2*	2*	C*	1*		
D^1	1*	2*	2*	C*	1*	0*	C*	C*	C*	1*		

							L1	L2	L3	L4	L5	L6
							1	3	0	1		
D^1	C*	2*	C*	C*	1*	1*	1*	C*	1*	1*		

							L1	L2	L3	L4	L5	L6
							1	3	1	C		
D^1	C*	2*	0*	0*	1*	1*	1*	1*	C*	1*		

							L1	L2	L3	L4	L5	L6
							2	1	0	2		
B^1	C*	C*	C*	C*	1*	2*	1*	C*	2*	1*		
C^1	C*	0*	C*	0*	1*	2*	1*	C*	2*	1*		
D^1	0*	C*	0*	0*	1*	2*	1*	C*	2*	1*		
D^1	2*	1*	C*	2*	1*	0*	0*	C*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							2	1	1	1		
1												
B	0*	0*	0*	0*	1*	2*	1*	1*	1*	1*		
1												
C	0*	0*	0*	0*	1*	2*	1*	1*	1*	1*		
1												
D	0*	0*	0*	0*	1*	2*	1*	1*	1*	1*		
1												
D	2*	0*	0*	0*	1*	0*	1*	1*	1*	1*		
1												
D	2*	1*	1*	1*	1*	0*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							2	1	2	0		
1												
B	0*	0*	0*	0*	1*	2*	1*	2*	0*	1*		
1												
C	2*	0*	0*	0*	1*	2*	1*	2*	0*	1*		
1												
D	0*	0*	0*	0*	1*	2*	1*	2*	0*	1*		
1												
D	2*	0*	0*	0*	1*	0*	1*	2*	0*	1*		
1												
D	2*	1*	2*	0*	1*	0*	0*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							2	2	0	1		
1												
D	0*	2*	0*	0*	1*	2*	0*	0*	1*	1*		

							L1	L2	L3	L4	L5	L6
							2	2	1	0		
1												
D	2*	0*	0*	0*	1*	0*	2*	1*	0*	1*		

							L1	L2	L3	L4	L5	L6
							2	3	0	0		
1												
D	2*	0*	0*	0*	1*	0*	3*	0*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                      L1  L2  L3  L4  L5  L6
                                       3   C   C  -2
     1
  D     2*  C*  0*  0*  1*  1*  C*  C*  2*  1*
-------------------------------------------------------------
                                      L1  L2  L3  L4  L5  L6
                                       3   C   1   1
     1
  D     2*  C*  C*  C*  1*  1*  0*  1*  1*  1*
-------------------------------------------------------------
                                      L1  L2  L3  L4  L5  L6
                                       3   1   0   1
     1
  D     2*  C*  C*  C*  1*  1*  1*  C*  1*  1*
-------------------------------------------------------------
                                      L1  L2  L3  L4  L5  L6
                                       3   1   1   0
     1
  D     2*  C*  0*  C*  1*  1*  1*  1*  C*  1*
-------------------------------------------------------------
                                      L1  L2  L3  L4  L5  L6
                                       3   2   C   C
     1
  D     C*  2*  0*  C*  1*  3*  C*  C*  C*  1*
-------------------------------------------------------------
                                      L1  L2  L3  L4  L5  L6
                                       4   C   C   1
     1
  D     2*  C*  0*  C*  1*  2*  C*  C*  1*  1*
-------------------------------------------------------------
                                      L1  L2  L3  L4  L5  L6
                                       5   C   C   C
     1
  D     2*  C*  C*  C*  1*  3*  C*  C*  0*  1*
-------------------------------------------------------------
                                      L1  L2  L3  L4  L5  L6
                                       0   2   2   2
     1
  D     C*  1*  1*  1*  1*  0*  1*  1*  1*  1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							0	2	3	1		
C^{1}	0*	1*	1*	1*	1*	0*	1*	2*	0*	1*		
D^{1}	0*	1*	2*	0*	1*	0*	1*	1*	1*	1*		
							L1	L2	L3	L4	L5	L6
							0	2	4	0		
C^{1}	0*	1*	2*	0*	1*	0*	1*	2*	0*	1*		
							L1	L2	L3	L4	L5	L6
							0	3	2	1		
C^{1}	0*	2*	1*	0*	1*	0*	1*	1*	1*	1*		
D^{1}	0*	1*	1*	1*	1*	0*	-2*	1*	0*	1*		
D^{2}	0*	2*	1*	0*	1*	0*	1*	1*	1*	1*		
							L1	L2	L3	L4	L5	L6
							0	3	3	0		
C^{1}	0*	2*	1*	0*	1*	0*	1*	2*	0*	1*		
D^{1}	0*	1*	2*	0*	1*	0*	2*	1*	0*	1*		
D^{2}	0*	2*	1*	0*	1*	0*	1*	2*	0*	1*		
							L1	L2	L3	L4	L5	L6
							0	4	1	1		
D^{1}	0*	1*	1*	1*	1*	0*	3*	0*	0*	1*		
D^{1}	0*	3*	0*	0*	1*	0*	1*	1*	1*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

								L1	L2	L3	L4	L5	L6
								0	4	2	5		
C	1	0*	2*	1*	0*	1*	0*	2*	1*	0*	1*		
D	1	0*	1*	2*	0*	1*	0*	3*	0*	0*	1*		
C	2	0*	2*	1*	0*	1*	0*	2*	1*	0*	1*		
D	1	0*	3*	0*	0*	1*	0*	1*	2*	0*	1*		

								L1	L2	L3	L4	L5	L6
								6	5	1	3		
C	1	0*	2*	1*	0*	1*	0*	3*	0*	0*	1*		
D	2	0*	2*	1*	0*	1*	0*	3*	0*	0*	1*		
~~D~~	1	~~0*~~	~~3*~~	~~0*~~	~~0*~~	~~1*~~	~~0*~~	~~2*~~	~~1*~~	~~0*~~	~~1*~~		

								L1	L2	L3	L4	L5	L6
								2	6	0	0		
D	1	0*	2*	0*	0*	1*	0*	3*	0*	0*	1*		

								L1	L2	L3	L4	L5	L6
								1	1	1	3		
D	1	0*	1*	1*	1*	1*	1*	0*	0*	2*	1*		
~~D~~	1	~~1*~~	~~0*~~	~~0*~~	~~2*~~	~~1*~~	~~0*~~	~~1*~~	~~1*~~	~~1*~~	~~1*~~		

								L1	L2	L3	L4	L5	L6
								1	1	2	2		
C	1	0*	1*	1*	1*	1*	1*	0*	1*	1*	1*		
C	1	0*	1*	2*	0*	1*	1*	0*	0*	2*	1*		
C	1	1*	0*	0*	2*	1*	0*	1*	2*	0*	1*		
D	1	1*	0*	1*	1*	1*	0*	1*	1*	1*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							1	1	3	1		
D^{1}	0*	1*	2*	0*	1*	1*	0*	1*	1*	1*		
D^{1}	1*	0*	1*	1*	1*	0*	1*	2*	0*	1*		

							L1	L2	L3	L4	L5	L6
							1	2	1	2		
C^{1}	0*	2*	1*	0*	1*	1*	0*	0*	2*	1*		
C^{1}	1*	1*	0*	1*	1*	0*	1*	1*	1*	1*		
D^{1}	0*	1*	1*	1*	1*	1*	1*	0*	1*	1*		
D^{2}	0*	2*	1*	0*	1*	1*	0*	0*	2*	1*		
D^{1}	1*	0*	0*	2*	1*	0*	2*	1*	0*	1*		
D^{1}	1*	1*	0*	1*	1*	0*	1*	1*	1*	1*		

							L1	L2	L3	L4	L5	L6
							1	2	2	1		
C^{1}	0*	2*	1*	0*	1*	1*	0*	1*	1*	1*		
C^{1}	1*	1*	0*	1*	1*	0*	1*	2*	0*	1*		
C^{1}	1*	1*	1*	0*	1*	0*	1*	1*	1*	1*		
D^{1}	0*	1*	1*	1*	1*	1*	1*	1*	0*	1*		
C^{1}	0*	1*	2*	0*	1*	1*	1*	0*	1*	1*		
D^{2}	0*	2*	1*	0*	1*	1*	0*	1*	1*	1*		
C^{1}	1*	0*	1*	1*	1*	0*	2*	1*	0*	1*		
D^{1}	1*	1*	0*	1*	1*	0*	1*	2*	0*	1*		
C^{1}	1*	1*	1*	0*	1*	0*	1*	1*	1*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

								L1	L2	L3	L4	L5	L6
								1	2	3	C		
C	1	1*	1*	1*	C*	1*	0*	1*	2*	C*	1*		
D	1	0*	1*	2*	C*	1*	1*	1*	1*	C*	1*		
D	1	1*	1*	1*	C*	1*	0*	1*	2*	0*	1*		

								L1	L2	L3	L4	L5	L6
								1	3	0	2		
B	1	C*	1*	C*	C*	1*	1*	2*	C*	2*	1*		
C	1	C*	1*	C*	0*	1*	1*	2*	C*	2*	1*		
D	1	C*	1*	C*	C*	1*	1*	2*	C*	2*	1*		
D	1	C*	3*	0*	0*	1*	1*	0*	C*	2*	1*		
D	1	1*	0*	0*	2*	1*	0*	3*	C*	0*	1*		
C	1	1*	2*	0*	2*	1*	0*	1*	C*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

								L1	L2	L3	L4	L5	L6
								1	3	1	1		
B	1	0*	1*	0*	0*	1*	1*	2*	1*	1*	1*		
C	1	0*	1*	0*	0*	1*	1*	2*	1*	1*	1*		
C	1	0*	2*	1*	0*	1*	1*	1*	0*	1*	1*		
C	1	1*	1*	0*	1*	1*	0*	2*	1*	0*	1*		
D	1	0*	1*	0*	0*	1*	1*	2*	1*	1*	1*		
D	2	0*	2*	1*	0*	1*	1*	1*	0*	1*	1*		
D	1	0*	3*	0*	0*	1*	1*	0*	1*	1*	1*		
D	1	1*	0*	1*	1*	1*	0*	3*	0*	0*	1*		
D	1	1*	1*	0*	1*	1*	0*	2*	1*	0*	1*		
D	1	1*	2*	0*	0*	1*	0*	1*	1*	1*	1*		
D	1	1*	2*	1*	1*	1*	0*	1*	0*	0*	1*		

								L1	L2	L3	L4	L5	L6
								1	3	2	0		
B	1	0*	1*	0*	0*	1*	1*	2*	2*	0*	1*		
C	1	0*	1*	0*	0*	1*	1*	2*	2*	0*	1*		
C	1	0*	2*	1*	0*	1*	1*	1*	1*	0*	1*		
C	1	1*	1*	1*	0*	1*	0*	2*	1*	0*	1*		
D	1	0*	1*	0*	0*	1*	1*	2*	2*	0*	1*		
D	2	0*	2*	1*	0*	1*	1*	1*	1*	0*	1*		
D	1	1*	1*	1*	0*	1*	0*	2*	1*	0*	1*		
D	1	1*	2*	0*	0*	1*	0*	1*	2*	0*	1*		
D	1	1*	2*	2*	0*	1*	0*	1*	0*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                          L1   L2   L3   L4   L5   L6
                                           1    4    0    1
      1
  C      1*   1*   0*   1*   1*   0*   3*   0*   0*   1*
      1
  C      0*   3*   0*   0*   1*   1*   1*   0*   1*   1*
      1
  D      1*   1*   0*   1*   1*   0*   3*   0*   0*   1*

                                          L1   L2   L3   L4   L5   L6
                                           1    4    1    0
      1
  C      1*   1*   1*   0*   1*   0*   3*   0*   0*   1*
      1
  D      0*   3*   0*   0*   1*   1*   1*   1*   0*   1*
      1
  D      1*   1*   1*   0*   1*   0*   3*   0*   0*   1*
      1
  D      1*   2*   0*   0*   1*   0*   2*   1*   0*   1*

                                          L1   L2   L3   L4   L5   L6
                                           1    5    0    0
      1
  D      1*   2*   0*   0*   1*   0*   3*   0*   0*   1*

                                          L1   L2   L3   L4   L5   L6
                                           2    0    0    4
      1
  D      1*   0*   0*   2*   1*   1*   0*   0*   2*   1*

                                          L1   L2   L3   L4   L5   L6
                                           2    0    1    3
      1
  C      1*   0*   0*   2*   1*   1*   0*   1*   1*   1*
      1
  C      1*   0*   1*   1*   1*   1*   0*   0*   2*   1*

                                          L1   L2   L3   L4   L5   L6
                                           2    0    2    2
      1
  C      1*   0*   1*   1*   1*   1*   0*   1*   1*   1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							2	1	0	3		
C^1	1*	1*	0*	1*	1*	1*	0*	0*	2*	1*		
D^1	1*	0*	0*	2*	1*	1*	1*	0*	1*	1*		
D^1	1*	1*	0*	1*	1*	1*	0*	0*	2*	1*		

							L1	L2	L3	L4	L5	L6
							2	1	1	2		
C^1	1*	1*	0*	1*	1*	1*	0*	1*	1*	1*		
C^1	1*	1*	1*	0*	1*	1*	0*	0*	2*	1*		
C^1	2*	0*	0*	1*	1*	0*	1*	1*	1*	1*		
D^1	0*	1*	1*	1*	1*	2*	0*	0*	1*	1*		
D^1	1*	0*	0*	2*	1*	1*	1*	1*	0*	1*		
D^1	1*	0*	1*	1*	1*	1*	1*	0*	1*	1*		
D^1	1*	1*	0*	1*	1*	1*	0*	1*	1*	1*		
D^1	1*	1*	1*	0*	1*	1*	0*	0*	2*	1*		
D^1	2*	0*	0*	1*	1*	0*	1*	1*	1*	1*		

							L1	L2	L3	L4	L5	L6
							2	1	2	1		
C^1	1*	1*	1*	0*	1*	1*	0*	1*	1*	1*		
C^1	2*	0*	0*	1*	1*	0*	1*	[illegible]	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							2	2	0	2		
B^1	0*	1*	0*	0*	1*	2*	1*	0*	2*	1*		
B^1	1*	0*	0*	0*	1*	1*	2*	0*	2*	1*		
C^1	0*	1*	0*	0*	1*	2*	1*	0*	2*	1*		
C^1	1*	0*	0*	0*	1*	1*	2*	0*	2*	1*		
C^1	1*	1*	0*	1*	1*	1*	1*	0*	1*	1*		
D^1	0*	1*	0*	0*	1*	2*	1*	0*	2*	1*		
D^1	1*	0*	0*	0*	1*	1*	2*	0*	2*	1*		
D^1	1*	1*	0*	1*	1*	1*	1*	0*	1*	1*		
D^1	1*	2*	0*	0*	1*	1*	0*	0*	2*	1*		
D^1	1*	2*	0*	2*	1*	1*	0*	0*	0*	1*		
D^1	2*	1*	0*	2*	1*	0*	1*	0*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

								L1	L2	L3	L4	L5	L6
								2	2	1	1		
B	1	0*	1*	0*	0*	1*	2*	1*	1*	1*	1*		
B	1	1*	0*	0*	0*	1*	1*	2*	1*	1*	1*		
C	1	0*	1*	0*	0*	1*	2*	1*	1*	1*	1*		
C	1	0*	2*	1*	0*	1*	2*	0*	0*	1*	1*		
C	1	1*	0*	0*	0*	1*	1*	2*	1*	1*	1*		
C	1	1*	1*	0*	1*	1*	1*	1*	1*	0*	1*		
C	1	1*	1*	1*	0*	1*	1*	1*	0*	1*	1*		
C	1	2*	0*	0*	1*	1*	0*	2*	1*	0*	1*		
D	1	0*	1*	0*	0*	1*	2*	1*	1*	1*	1*		
D	2	0*	2*	1*	0*	1*	2*	0*	0*	1*	1*		
D	1	1*	0*	0*	0*	1*	1*	2*	1*	1*	1*		
D	1	1*	1*	0*	1*	1*	1*	1*	1*	0*	1*		
D	1	1*	1*	1*	0*	1*	1*	1*	0*	1*	1*		
D	1	1*	2*	0*	0*	1*	1*	0*	1*	1*	1*		
D	1	1*	2*	1*	1*	1*	1*	0*	0*	0*	1*		
D	1	2*	0*	0*	1*	1*	0*	2*	1*	0*	1*		
D	2	2*	1*	0*	0*	1*	0*	1*	1*	1*	1*		
D	1	2*	1*	1*	1*	1*	0*	1*	0*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							2	2	2	0		
B^{1}	0*	1*	0*	0*	1*	2*	1*	2*	0*	1*		
B^{1}	1*	0*	0*	0*	1*	1*	2*	2*	0*	1*		
C^{1}	0*	1*	0*	0*	1*	2*	1*	2*	0*	1*		
C^{1}	1*	0*	0*	0*	1*	1*	2*	2*	0*	1*		
C^{1}	1*	1*	1*	0*	1*	1*	1*	1*	0*	1*		
D^{1}	0*	1*	0*	0*	1*	2*	1*	2*	0*	1*		
D^{1}	1*	0*	0*	0*	1*	1*	2*	2*	0*	1*		
D^{1}	1*	1*	1*	0*	1*	1*	1*	1*	0*	1*		
D^{1}	1*	2*	2*	0*	1*	1*	0*	0*	0*	1*		
D^{2}	2*	1*	0*	0*	1*	0*	1*	2*	0*	1*		
D^{1}	2*	1*	2*	0*	1*	0*	1*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							2	3	0	1		
C^{1}	2*	0*	0*	1*	1*	0*	3*	0*	0*	1*		
D^{1}	0*	3*	0*	0*	1*	2*	0*	0*	1*	1*		
D^{1}	1*	2*	0*	0*	1*	1*	1*	0*	1*	1*		
D^{1}	2*	0*	0*	1*	1*	0*	3*	0*	0*	1*		

							L1	L2	L3	L4	L5	L6
							2	3	1	0		
D^{1}	1*	2*	0*	0*	1*	1*	1*	1*	0*	1*		
D^{2}	2*	1*	0*	0*	1*	0*	2*	1*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                        L1  L2  L3  L4  L5  L6
                                         2   4   C   C
       2
   D     2*  1*  C*  0*  1*  0*  3*  C*  C*  1*

                                        L1  L2  L3  L4  L5  L6
                                         3   C   C   3
       1
   C     2*  0*  0*  1*  1*  1*  C*  C*  2*  1*
       1
   C     1*  C*  C*  2*  1*  2*  C*  C*  1*  1*
       1
   C     2*  C*  0*  1*  1*  1*  C*  C*  2*  1*

                                        L1  L2  L3  L4  L5  L6
                                         3   C   1   2
       1
   C     2*  C*  C*  1*  1*  1*  C*  1*  1*  1*
       1
   C     1*  C*  1*  1*  1*  2*  C*  C*  1*  1*
       1
   C     2*  C*  0*  1*  1*  1*  C*  1*  1*  1*

                                        L1  L2  L3  L4  L5  L6
                                         3   1   C   2
       1
   B     1*  C*  C*  0*  1*  2*  1*  C*  2*  1*
       1
   C     1*  C*  C*  C*  1*  2*  1*  C*  2*  1*
       1
   C     1*  1*  C*  1*  1*  2*  C*  C*  1*  1*
       1
   C     2*  C*  C*  1*  1*  1*  1*  C*  1*  1*
       1
   D     1*  C*  C*  C*  1*  2*  1*  C*  2*  1*
       1
   D     1*  1*  C*  1*  1*  2*  0*  C*  1*  1*
       1
   D     2*  C*  C*  1*  1*  1*  1*  C*  1*  1*
       2
   D     2*  1*  C*  C*  1*  1*  C*  C*  2*  1*
       1
   D     2*  1*  0*  2*  1*  1*  C*  C*  C*  1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

								L1	L2	L3	L4	L5	L6
								3	1	1	1		
B	1	1*	0*	0*	0*	1*	2*	1*	1*	1*	1*		
C	1	1*	0*	0*	0*	1*	2*	1*	1*	1*	1*		
C	1	1*	1*	1*	0*	1*	2*	0*	0*	1*	1*		
C	1	2*	0*	0*	1*	1*	1*	1*	1*	0*	1*		
D	1	0*	1*	1*	1*	1*	3*	0*	0*	0*	1*		
D	1	1*	0*	0*	0*	1*	2*	1*	1*	1*	1*		
D	1	1*	1*	1*	0*	1*	2*	0*	0*	1*	1*		
D	1	2*	0*	0*	1*	1*	1*	1*	1*	0*	1*		
D	2	2*	1*	0*	0*	1*	1*	0*	1*	1*	1*		
D	1	2*	1*	1*	1*	1*	1*	0*	0*	0*	1*		
D	1	3*	0*	0*	0*	1*	0*	1*	1*	1*	1*		

								L1	L2	L3	L4	L5	L6
								3	1	2	0		
B	1	1*	0*	0*	0*	1*	2*	1*	2*	0*	1*		
C	1	1*	0*	0*	0*	1*	2*	1*	2*	0*	1*		
~~D~~	1	~~0*~~	~~1*~~	~~2*~~	~~0*~~	~~1*~~	~~3*~~	~~0*~~	~~0*~~	~~0*~~	~~1*~~		
D	1	1*	0*	0*	0*	1*	2*	1*	2*	0*	1*		
D	1	2*	1*	2*	0*	1*	1*	0*	0*	0*	1*		
~~D~~	1	~~3*~~	~~0*~~	~~0*~~	~~0*~~	~~1*~~	~~3*~~	~~1*~~	~~2*~~	~~0*~~	~~1*~~		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                                  L1  L2  L3  L4  L5  L6
                                                   3   2   C   1
        1
   D    1*  2*  0*  C*  1*  2*  C*  C*  1*  1*
        2
   D    2*  1*  C*  0*  1*  1*  1*  C*  1*  1*

                                                  L1  L2  L3  L4  L5  L6
                                                   3   2   1   C
        1
   C    C*  2*  1*  0*  1*  3*  C*  C*  0*  1*
        2
   D    C*  2*  1*  C*  1*  3*  C*  C*  C*  1*
        2
   D    2*  1*  C*  0*  1*  1*  1*  1*  C*  1*
        1
   D    3*  C*  C*  C*  1*  0*  2*  1*  C*  1*

                                                  L1  L2  L3  L4  L5  L6
                                                   3   3   0   C
        1
   D    C*  3*  0*  C*  1*  3*  C*  C*  C*  1*
        1
   D    3*  C*  C*  0*  1*  0*  3*  C*  C*  1*

                                                  L1  L2  L3  L4  L5  L6
                                                   4   C   0   2
        1
   C    2*  C*  0*  1*  1*  2*  C*  C*  1*  1*
        1
   D    1*  C*  C*  2*  1*  3*  C*  C*  C*  1*
        1
   D    2*  C*  0*  1*  1*  2*  C*  C*  1*  1*
        1
   D    ?*  C*  C*  C*  1*  1*  C*  C*  2*  1*

                                                  L1  L2  L3  L4  L5  L6
                                                   4   C   1   1
        1
   D    1*  C*  1*  1*  1*  3*  C*  C*  C*  1*
        1
   D    3*  C*  C*  C*  1*  1*  C*  1*  1*  1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

	L1	L2	L3	L4	L5	L6
	4	1	0	1		

C 1	1*	1*	0*	1*	1*	3*	0*	C*	C*	1*
D 1	1*	1*	0*	1*	1*	3*	0*	C*	C*	1*
D 2	2*	1*	C*	0*	1*	2*	0*	C*	1*	1*
D 1	3*	C*	0*	0*	1*	1*	1*	C*	1*	1*

	L1	L2	L3	L4	L5	L6
	4	1	1	C		

C 1	1*	1*	1*	0*	1*	3*	0*	C*	C*	1*
D 1	1*	1*	1*	0*	1*	3*	C*	C*	C*	1*
D 1	3*	C*	0*	C*	1*	1*	1*	1*	0*	1*

	L1	L2	L3	L4	L5	L6
	4	2	C	C		

D 1	1*	2*	0*	0*	1*	3*	0*	C*	C*	1*

	L1	L2	L3	L4	L5	L6
	5	C	C	1		

C 1	2*	C*	0*	1*	1*	3*	0*	C*	C*	1*
D 1	2*	C*	0*	1*	1*	3*	C*	0*	0*	1*
C 1	3*	C*	C*	0*	1*	2*	C*	C*	1*	1*

	L1	L2	L3	L4	L5	L6
	5	1	C	C		

D 2	2*	1*	0*	0*	1*	3*	0*	C*	0*	1*

	L1	L2	L3	L4	L5	L6
	6	C	C	0		

D 1	3*	0*	C*	0*	1*	3*	0*	C*	C*	1*

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                          L1  L2  L3  L4  L5  L6
                                           1   4   0   2
      1
   D     0*  2*  0*  0*  1*  1*  2*  0*  2*  1*

                                          L1  L2  L3  L4  L5  L6
                                           1   4   1   1
      1
   D     0*  2*  0*  0*  1*  1*  2*  1*  1*  1*

                                          L1  L2  L3  L4  L5  L6
                                           1   4   2   0
      1
   D     0*  2*  0*  0*  1*  1*  2*  2*  0*  1*

                                          L1  L2  L3  L4  L5  L6
                                           2   3   0   2
      1
   D     0*  2*  0*  0*  1*  2*  1*  0*  2*  1*

                                          L1  L2  L3  L4  L5  L6
                                           2   3   1   1
      1
   D     0*  2*  0*  0*  1*  2*  1*  1*  1*  1*

                                          L1  L2  L3  L4  L5  L6
                                           2   3   2   0
      1
   D     0*  2*  0*  0*  1*  2*  1*  2*  0*  1*

                                          L1  L2  L3  L4  L5  L6
                                           3   2   0   2
      1
   D     2*  0*  0*  0*  1*  1*  2*  0*  2*  1*

                                          L1  L2  L3  L4  L5  L6
                                           3   2   1   1
      1
   D     2*  0*  0*  0*  1*  1*  2*  1*  1*  1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                         L1  L2  L3  L4  L5  L6
                                          3   2  -2   0
      1
   D     2*  0*  0*  0*  1*  1*  2*  2*  0*  1*

                                         L1  L2  L3  L4  L5  L6
                                          4   1   0   2
      1
   D     2*  0*  0*  0*  1*  2*  1*  0*  2*  1*

                                         L1  L2  L3  L4  L5  L6
                                          4   1   1   1
      1
   D     2*  0*  0*  0*  1*  2*  1*  1*  1*  1*

                                         L1  L2  L3  L4  L5  L6
                                          4   1   2   0
      1
   D     2*  0*  0*  0*  1*  2*  1*  2*  0*  1*

                                         L1  L2  L3  L4  L5  L6
                                          1   3   1   0
      1
   D     0*  1*  1*  1*  1*  1*  2*  0*  2*  1*
      1
   D     1*  2*  0*  2*  1*  0*  1*  1*  1*  1*

                                         L1  L2  L3  L4  L5  L6
                                          1   3   2   2
      1
   D     0*  1*  1*  1*  1*  1*  2*  1*  1*  1*
      1
   D     0*  1*  2*  0*  1*  1*  2*  0*  2*  1*
      1
   D     1*  2*  0*  2*  1*  0*  1*  2*  0*  1*
      1
   D     1*  2*  1*  1*  1*  0*  1*  1*  1*  1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

								L1	L2	L3	L4	L5	L6
								1	2	3	1		
D	1	0*	1*	1*	1*	1*	1*	2*	2*	0*	1*		
D	1	0*	1*	2*	0*	1*	1*	2*	1*	1*	1*		
D	1	1*	2*	1*	1*	1*	0*	1*	2*	0*	1*		
D	1	1*	2*	2*	0*	1*	0*	1*	1*	1*	1*		

								L1	L2	L3	L4	L5	L6
								1	3	4	0		
D	1	0*	1*	2*	0*	1*	1*	2*	2*	0*	1*		
D	1	1*	2*	2*	0*	1*	0*	1*	2*	0*	1*		

								L1	L2	L3	L4	L5	L6
								1	4	1	2		
C	1	0*	2*	1*	0*	1*	1*	2*	0*	2*	1*		
D	2	0*	2*	1*	0*	1*	1*	2*	0*	2*	1*		
D	1	1*	2*	0*	2*	1*	0*	2*	1*	0*	1*		

								L1	L2	L3	L4	L5	L6
								1	4	2	1		
C	1	0*	2*	1*	0*	1*	1*	2*	1*	1*	1*		
D	2	0*	2*	1*	0*	1*	1*	2*	1*	1*	1*		
D	1	1*	2*	1*	1*	1*	0*	2*	1*	0*	1*		

								L1	L2	L3	L4	L5	L6
								1	4	3	0		
C	1	0*	2*	1*	0*	1*	1*	2*	2*	0*	1*		
D	2	0*	2*	1*	0*	1*	1*	2*	2*	0*	1*		
D	1	1*	2*	2*	0*	1*	0*	2*	1*	0*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                         L1   L2   L3   L4   L5   L6
                                          1    5    0    2
      1
D        0*   3*   0*   0*   1*   1*   2*   0*   2*   1*
      1
D        1*   2*   0*   2*   1*   0*   3*   0*   0*   1*
                                         L1   L2   L3   L4   L5   L6
                                          1    5    1    1
      1
D        0*   3*   0*   0*   1*   1*   2*   1*   1*   1*
      1
D        1*   2*   1*   1*   1*   0*   3*   0*   0*   1*
                                         L1   L2   L3   L4   L5   L6
                                          1    5    2    0
      1
D        0*   3*   0*   0*   1*   1*   2*   2*   0*   1*
      1
D        1*   2*   2*   0*   1*   0*   3*   0*   0*   1*
                                         L1   L2   L3   L4   L5   L6
                                          2    2    0    4
      1
D        1*   0*   0*   2*   1*   1*   2*   0*   2*   1*
      1
D        1*   2*   0*   2*   1*   1*   0*   0*   2*   1*
                                         L1   L2   L3   L4   L5   L6
                                          2    2    1    3
      1
D        0*   1*   1*   1*   1*   2*   1*   0*   2*   1*
      1
D        1*   0*   0*   2*   1*   1*   2*   1*   1*   1*
      1
D        1*   0*   1*   1*   1*   1*   2*   0*   2*   1*
      1
D        1*   2*   0*   2*   1*   1*   0*   1*   1*   1*
      1
D        1*   2*   1*   1*   1*   1*   0*   0*   2*   1*
      1
D        2*   1*   0*   2*   1*   0*   1*   1*   1*   1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1 2	L2 2	L3 2	L4 2	L5	L6
~~D_1~~	~~3*~~	~~1*~~	~~1*~~	~~1*~~	~~1*~~	~~2*~~	~~1*~~	~~1*~~	~~1*~~	~~1*~~		
D_1	0*	1*	2*	0*	1*	2*	1*	0*	2*	1*		
D_1	1*	0*	0*	2*	1*	1*	2*	2*	0*	1*		
~~D_1~~	~~1*~~	~~0*~~	~~1*~~	~~1*~~	~~1*~~	~~1*~~	~~2*~~	~~1*~~	~~1*~~	~~1*~~		
D_1	1*	2*	1*	1*	1*	1*	0*	1*	1*	1*		
D_1	1*	2*	2*	0*	1*	1*	0*	0*	2*	1*		
~~D_1~~	~~2*~~	~~1*~~	~~0*~~	~~2*~~	~~1*~~	~~0*~~	~~1*~~	~~2*~~	~~0*~~	~~1*~~		
D_1	2*	1*	1*	1*	1*	0*	1*	1*	1*	1*		

							L1 2	L2 2	L3 3	L4 1	L5	L6
D_1	0*	1*	1*	1*	1*	2*	1*	2*	0*	1*		
D_1	0*	1*	2*	0*	1*	2*	1*	1*	1*	1*		
D_1	1*	0*	1*	1*	1*	1*	2*	2*	0*	1*		
D_1	1*	2*	2*	0*	1*	1*	0*	1*	1*	1*		
D_1	2*	1*	1*	1*	1*	0*	1*	2*	0*	1*		
D_1	2*	1*	2*	0*	1*	0*	1*	1*	1*	1*		

							L1 2	L2 2	L3 4	L4 0	L5	L6
D_1	0*	1*	2*	0*	1*	2*	1*	2*	0*	1*		
~~D_1~~	~~2*~~	~~1*~~	~~2*~~	~~0*~~	~~1*~~	~~0*~~	~~1*~~	~~2*~~	~~0*~~	~~1*~~		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                          L1  L2  L3  L4  L5  L6
                                           2   3   C   3
       1
    C     1*  1*  0*  1*  1*  1*  2*  C*  2*  1*
       1
    C     1*  1*  C*  1*  1*  1*  2*  C*  2*  1*
       1
    C     1*  2*  0*  2*  1*  1*  1*  C*  1*  1*

                                          L1  L2  L3  L4  L5  L6
                                           2   3   1   2
       1
    C     C*  2*  1*  C*  1*  2*  1*  C*  2*  1*
       1
    C     1*  1*  C*  1*  1*  1*  2*  1*  1*  1*
       1
    C     1*  1*  1*  C*  1*  1*  2*  C*  2*  1*
       2
    C     C*  2*  1*  C*  1*  2*  1*  C*  2*  1*
       1
    C     1*  1*  0*  1*  1*  1*  2*  1*  1*  1*
       1
    C     1*  1*  1*  C*  1*  1*  2*  C*  2*  1*
       1
    C     1*  2*  0*  2*  1*  1*  1*  1*  C*  1*
       1
    C     1*  2*  1*  1*  1*  1*  1*  C*  1*  1*
       1
    C     2*  1*  0*  2*  1*  0*  2*  1*  C*  1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

	L1	L2	L3	L4	L5	L6
	2	3	2	1		

1	C	0*	2*	1*	0*	1*	2*	1*	1*	1*	1*
1	C	1*	1*	0*	1*	1*	1*	2*	2*	0*	1*
1	C	1*	1*	1*	0*	1*	1*	2*	1*	1*	1*
2	D	0*	2*	1*	0*	1*	2*	1*	1*	1*	1*
1	D	1*	1*	0*	1*	1*	1*	2*	2*	0*	1*
1	D	1*	1*	1*	0*	1*	1*	2*	1*	1*	1*
1	D	1*	2*	1*	1*	1*	1*	1*	1*	0*	1*
1	D	1*	2*	2*	0*	1*	1*	1*	0*	1*	1*
1	D	2*	1*	1*	1*	1*	0*	2*	1*	0*	1*

	L1	L2	L3	L4	L5	L6
	2	3	3	0		

1	C	0*	2*	1*	0*	1*	2*	1*	2*	0*	1*
1	C	1*	1*	1*	0*	1*	1*	2*	2*	0*	1*
2	~~D~~	~~0*~~	~~2*~~	~~1*~~	~~0*~~	~~1*~~	~~2*~~	~~1*~~	~~2*~~	~~0*~~	~~1*~~
1	D	1*	1*	1*	0*	1*	1*	2*	2*	0*	1*
1	D	1*	2*	2*	0*	1*	1*	1*	1*	0*	1*
1	~~D~~	~~2*~~	~~1*~~	~~2*~~	~~0*~~	~~1*~~	~~0*~~	~~2*~~	~~1*~~	~~0*~~	~~1*~~

	L1	L2	L3	L4	L5	L6
	2	4	0	2		

1	C	0*	3*	0*	0*	1*	2*	1*	0*	2*	1*
1	D	1*	2*	0*	0*	1*	1*	2*	0*	2*	1*
1	D	2*	1*	3*	2*	1*	0*	3*	0*	0*	1*

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                          L1   L2   L3   L4   L5   L6
                                          2    4    1    1

     1
  D     0*   3*   0*   0*   1*   2*   1*   1*   1*   1*
     1
  D     1*   2*   0*   0*   1*   1*   2*   1*   1*   1*
     1
  D     2*   1*   1*   1*   1*   0*   3*   0*   0*   1*

                                          L1   L2   L3   L4   L5   L6
                                          2    4    2    0
     1
  D     0*   3*   0*   0*   1*   2*   1*   2*   0*   1*
     1
  D     1*   2*   0*   0*   1*   1*   2*   2*   0*   1*
     1
  D     2*   1*   2*   0*   1*   0*   3*   0*   0*   1*

                                          L1   L2   L3   L4   L5   L6
                                          3    1    0    4
     1
  D     1*   0*   0*   2*   1*   2*   1*   0*   2*   1*
     1
  D     2*   1*   0*   2*   1*   1*   0*   0*   2*   1*

                                          L1   L2   L3   L4   L5   L6
                                          3    1    1    3
     1
  D     1*   0*   0*   2*   1*   2*   1*   1*   1*   1*
     1
  D     1*   0*   1*   1*   1*   2*   1*   0*   2*   1*
     1
  D     2*   1*   0*   2*   1*   1*   0*   1*   1*   1*
     1
  D     2*   1*   1*   1*   1*   1*   0*   0*   2*   1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							3	1	2	2		
D^1	1*	C*	C*	2*	1*	2*	1*	2*	0*	1*		
C^1	1*	C*	1*	1*	1*	2*	1*	1*	1*	1*		
~~C^1~~	~~2*~~	~~1*~~	~~1*~~	~~1*~~	~~1*~~	~~1*~~	~~0*~~	~~1*~~	~~1*~~	~~1*~~		
D^1	2*	1*	2*	C*	1*	1*	0*	C*	2*	1*		

							L1	L2	L3	L4	L5	L6
							3	1	3	1		
C^1	1*	C*	1*	1*	1*	2*	1*	2*	0*	1*		
C^1	2*	1*	2*	C*	1*	1*	0*	1*	1*	1*		

							L1	L2	L3	L4	L5	L6
							3	2	C	3		
C^1	1*	1*	0*	1*	1*	2*	1*	C*	2*	1*		
C^1	2*	C*	C*	1*	1*	1*	2*	C*	2*	1*		
~~C^1~~	~~1*~~	~~1*~~	~~0*~~	~~1*~~	1*	2*	~~1*~~	C*	2*	1*		
D^1	1*	2*	0*	2*	1*	2*	0*	C*	1*	1*		
D^1	2*	C*	C*	1*	1*	1*	2*	C*	2*	1*		
~~D^1~~	~~2*~~	~~1*~~	~~C*~~	~~2*~~	~~1*~~	1*	1*	C*	1*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

							L1	L2	L3	L4	L5	L6
							3	2	1	2		
C_1	1*	1*	0*	1*	1*	2*	1*	1*	1*	1*		
C_1	1*	1*	1*	0*	1*	2*	1*	0*	2*	1*		
C_1	2*	0*	0*	1*	1*	1*	2*	1*	1*	1*		
D_1	1*	1*	0*	1*	1*	2*	1*	1*	1*	1*		
D_1	1*	1*	1*	0*	1*	2*	1*	0*	2*	1*		
D_1	1*	2*	1*	1*	1*	2*	0*	0*	1*	1*		
D_1	2*	0*	0*	1*	1*	1*	2*	1*	1*	1*		
D_1	2*	1*	0*	2*	1*	1*	1*	1*	0*	1*		
D_1	2*	1*	1*	1*	1*	1*	1*	0*	1*	1*		

							L1	L2	L3	L4	L5	L6
							3	2	2	1		
C_1	1*	1*	0*	1*	1*	2*	1*	2*	0*	1*		
C_1	1*	1*	1*	0*	1*	2*	1*	1*	1*	1*		
~~C_1~~	~~2*~~	~~0*~~	~~0*~~	~~1*~~	~~1*~~	~~1*~~	~~2*~~	~~2*~~	~~0*~~	1*		
D_1	1*	1*	0*	1*	1*	2*	1*	2*	0*	1*		
D_1	1*	1*	1*	0*	1*	2*	1*	1*	1*	1*		
C_1	1*	2*	2*	0*	1*	2*	0*	0*	1*	1*		
D_1	2*	0*	0*	1*	1*	1*	2*	2*	0*	1*		
D_1	2*	1*	1*	1*	1*	1*	1*	1*	0*	1*		
~~D_1~~	~~2*~~	~~1*~~	~~2*~~	~~0*~~	~~1*~~	1*	1*	0*	1*	1*		

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                         L1   L2   L3   L4   L5   L6
                                          3    2    3    0

       1
C        1*   1*   1*   0*   1*   2*   1*   2*   0*   1*
       1
D        1*   1*   1*   0*   1*   2*   1*   2*   0*   1*
       1
D        2*   1*   2*   0*   1*   1*   1*   1*   0*   1*

                                         L1   L2   L3   L4   L5   L6
                                          3    3    0    2
       1
D        1*   2*   0*   0*   1*   2*   1*   0*   2*   1*
       2
D        2*   1*   0*   0*   1*   1*   2*   0*   2*   1*

                                         L1   L2   L3   L4   L5   L6
                                          3    3    1    1
       1
D        1*   2*   0*   0*   1*   2*   1*   1*   1*   1*
       2
D        2*   1*   0*   0*   1*   1*   2*   1*   1*   1*

                                         L1   L2   L3   L4   L5   L6
                                          3    3    2    0
       1
C        1*   2*   0*   0*   1*   2*   1*   2*   0*   1*
       2
D        2*   1*   0*   0*   1*   1*   2*   2*   0*   1*

                                         L1   L2   L3   L4   L5   L6
                                          4    1    0    3
       1
C        2*   0*   0*   1*   1*   2*   1*   0*   2*   1*
       1
C        2*   0*   0*   1*   1*   2*   1*   0*   2*   1*
       1
D        2*   1*   0*   2*   1*   2*   0*   0*   1*   1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                                  L1   L2   L3   L4   L5   L6
                                                   4    1    1    2
       1
   C     2*   0*   0*   1*   1*   2*   1*   1*   1*   1*
       1
   D     2*   0*   0*   1*   1*   2*   1*   1*   1*   1*
       1
   D     2*   1*   1*   1*   1*   2*   0*   0*   1*   1*

                                                  L1   L2   L3   L4   L5   L6
                                                   4    1    2    1
       1
   C     2*   0*   0*   1*   1*   2*   1*   2*   0*   1*
       1
   D     2*   0*   0*   1*   1*   2*   1*   2*   0*   1*
       1
   D     2*   1*   2*   0*   1*   2*   0*   0*   1*   1*

                                                  L1   L2   L3   L4   L5   L6
                                                   4    2    0    2
       1
   D     1*   2*   0*   2*   1*   3*   0*   0*   0*   1*
       2
   D     2*   1*   0*   0*   1*   2*   1*   0*   2*   1*
       1
   D     2*   0*   0*   0*   1*   1*   2*   0*   2*   1*

                                                  L1   L2   L3   L4   L5   L6
                                                   4    2    1    1
       1
   D     1*   2*   1*   1*   1*   3*   0*   0*   0*   1*
       2
   D     2*   1*   0*   0*   1*   2*   1*   1*   1*   1*
       1
   D     3*   0*   0*   0*   1*   1*   2*   1*   1*   1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

L1	L2	L3	L4	L5	L6
4	2	2	0		

C_1	1*	2*	2*	0*	1*	3*	0*	0*	0*	1*
C_2	2*	1*	0*	0*	1*	2*	1*	2*	0*	1*
C_1	2*	0*	0*	0*	1*	1*	2*	2*	0*	1*

L1	L2	L3	L4	L5	L6
5	1	0	2		

C_1	2*	1*	0*	2*	1*	3*	0*	0*	0*	1*
C_1	3*	0*	0*	0*	1*	2*	1*	0*	2*	1*

L1	L2	L3	L4	L5	L6
5	1	1	1		

C_1	2*	1*	1*	1*	1*	3*	0*	0*	0*	1*
C_1	[illegible]	0*	0*	0*	1*	2*	1*	1*	1*	1*

L1	L2	L3	L4	L5	L6
5	1	2	0		

C_1	2*	1*	2*	0*	1*	3*	0*	0*	0*	1*
C_1	3*	0*	0*	0*	1*	2*	1*	2*	0*	1*

L1	L2	L3	L4	L5	L6
2	4	0	4		

C_1	1*	2*	0*	2*	1*	1*	2*	0*	2*	1*

L1	L2	L3	L4	L5	L6
2	4	1	3		

C_1	1*	2*	0*	2*	1*	1*	2*	1*	1*	1*
C_1	1*	2*	1*	1*	1*	1*	2*	0*	2*	1*

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

```
                                   L1   L2   L3   L4   L5   L6
                                   2    4    2    2
   1
 D    1*   2*   0*   2*   1*   1*   2*   2*   0*   1*
   1
 D    1*   2*   1*   1*   1*   1*   2*   1*   1*   1*
   1
 D    1*   2*   2*   0*   1*   1*   2*   0*   2*   1*

                                   L1   L2   L3   L4   L5   L6
                                   2    4    3    1
   1
 D    1*   2*   1*   1*   1*   1*   2*   2*   0*   1*
   1
 D    1*   2*   2*   0*   1*   1*   2*   1*   1*   1*

                                   L1   L2   L3   L4   L5   L6
                                   2    4    4    0
   1
 D    1*   2*   2*   0*   1*   1*   2*   2*   0*   1*

                                   L1   L2   L3   L4   L5   L6
                                   3    3    0    4
   1
 D    1*   2*   0*   2*   1*   2*   1*   0*   2*   1*
   1
 D    2*   1*   0*   2*   1*   1*   2*   0*   2*   1*

                                   L1   L2   L3   L4   L5   L6
                                   3    3    1    3
   1
 D    1*   2*   0*   2*   1*   2*   1*   1*   1*   1*
   1
 D    1*   2*   1*   1*   1*   2*   1*   0*   2*   1*
   1
 D    2*   1*   0*   2*   1*   1*   2*   1*   1*   1*
   1
 D    2*   1*   1*   1*   1*   1*   2*   0*   2*   1*
```

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (continued)

L1	L2	L3	L4	L5	L6
3	3	2	2		

D^1	1*	2*	0*	2*	1*	2*	1*	2*	0*	1*
D^1	1*	2*	1*	1*	1*	2*	1*	1*	1*	1*
D^1	1*	2*	2*	0*	1*	2*	1*	0*	2*	1*
D^1	2*	1*	0*	2*	1*	1*	2*	2*	0*	1*
D^1	2*	1*	1*	1*	1*	1*	2*	1*	1*	1*
D^1	2*	1*	2*	0*	1*	1*	2*	0*	2*	1*

L1	L2	L3	L4	L5	L6
3	3	3	1		

D^1	1*	2*	1*	1*	1*	2*	1*	2*	0*	1*
D^1	1*	2*	2*	0*	1*	2*	1*	1*	1*	1*
D^1	2*	1*	1*	1*	1*	1*	2*	2*	0*	1*
D^1	2*	1*	2*	0*	1*	1*	2*	1*	1*	1*

L1	L2	L3	L4	L5	L6
3	3	4	0		

D^1	1*	2*	2*	0*	1*	2*	1*	2*	0*	1*
D^1	2*	1*	2*	0*	1*	1*	2*	2*	0*	1*

L1	L2	L3	L4	L5	L6
4	2	0	4		

D^1	2*	1*	0*	2*	1*	2*	1*	0*	2*	1*

L1	L2	L3	L4	L5	L6
4	2	1	3		

D^1	2*	1*	0*	2*	1*	2*	1*	1*	1*	1*
D^1	2*	1*	1*	1*	1*	2*	1*	0*	2*	1*

Table 2-3. SERIES FOR ORTHOGONALITY CONDITION (concluded)

```
                                              L1   L2   L3   L4   L5   L6
                                               2    2    2    2
    1
  D    2*   1*   0*   2*   1*   2*   1*   2*   0*   1*
    1
  D    2*   1*   1*   1*   1*   2*   1*   1*   1*   1*
    1
  D    2*   1*   2*   0*   1*   2*   1*   0*   2*   1*

                                              L1   L2   L3   L4   L5   L6
                                               4    2    3    1
    1
  D    2*   1*   1*   1*   1*   2*   1*   2*   0*   1*
    1
  D    2*   1*   2*   0*   1*   2*   1*   1*   1*   1*

                                              L1   L2   L3   L4   L5   L6
                                               4    2    4    0
    1
  D    2*   1*   2*   0*   1*   2*   1*   2*   0*   1*

  ****************************************************************

  SAVE TAPE
  CASE ID=   X*L+Y*L
```

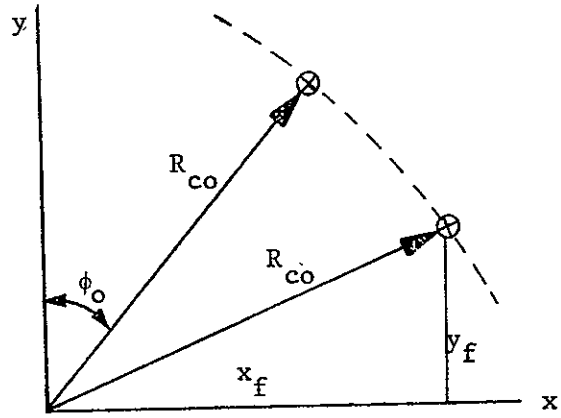
SECTION III

OPTIMAL RENDEZVOUS WITH SATELLITE IN CIRCULAR ORBIT

3.1 INTRODUCTION

The mathematical model for this problem is nearly the same as that for optimal ascent to circular orbit. The only difference is that one of the terminal conditions is different. Instead of a non-trivial transversality condition, there is a condition that describes the rendezvous of the powered vehicle and a target satellite.

The coordinate system is shown in the following figure.



We define:

$\dot{\theta}$ as the angular rate of the target

ϕ_o as the angle between launch vertical and the target at the time of "ignition". (Assumed to be known.)

t_{ig} as the time when the flight begins, the "ignition" time

t_o as the current initial time

t_f as the time from t_o till rendezvous

Δt as $(t_f - t_o)$.

At any t_o we compute ϕ as

$$\phi = \phi_o + \dot{\theta} (t_o - t_{ig}).$$

For the "rendezvous" terminal condition we have

$$x_f - R_{co} \text{ Sin } (\phi_o + \dot{\theta}\Delta t) = 0 ,$$

or $$y_f - R_{co} \text{ Cos } (\phi_o + \dot{\theta}\Delta t) = 0 .$$

In addition we have the terminal conditions for the circular orbit

$$V_f^2 - V_{co}^2 = 0$$

$$R_f^2 - R_{co}^2 = 0$$

$$\bar{R}_f \cdot \bar{V}_f = 0 .$$

The solution of this problem proceeds in the same way as that of the optimal ascent to circular orbit. The solution for Δt that was obtained in that problem applies here.

The expression of the rendezvous condition is written as

$$0 \approx W_o + W_1 \Delta t + W_2 \Delta t^2 + \cdots ,$$

where

$$W_o = x_o - R_{co} \sin (\phi_o + \dot{\theta}\Delta t)$$

$$W_1 = u_o$$

$$W_2 = \dot{u}_o$$

$$W_3 = \ddot{u}_o$$

This expansion can be put in terms of the multipliers by substitution for the u time derivatives. Together with similar expansions for the final radius and path angle conditions and the scaling conditions, we obtain a system of four algebraic equations in the multipliers.

The unknown parameter, time-to-cutoff, or Δt, appears implicitly in the rendezvous condition. It was necessary to approximate the sine term with a series expansion in order to have Δt appear explicitly. The term W_o is replaced by:

$$W_o \approx x_o - R_{co} (\phi_o + \dot{\theta}\Delta t - \phi_o^3 - 3\phi_o^2 \dot{\theta}\Delta t$$
$$- \phi_o \dot{\theta}^2 \Delta t^2 - \dot{\theta}^3 \Delta t^3).$$

3.2 SOLUTION FOR THE MULTIPLIERS

The solution for the multipliers in this problem was carried out in the same way as in the ascent-to-circular-orbit problem with the exception that the transversality condition of the circular problem is replaced by the rendezvous condition. To obtain a reasonably accurate solution for the multipliers it was necessary to have a fifth-order series representation for the rendezvous condition. The other conditions were represented by third-, fourth-, and fifth-order series.

Typical solutions are shown in Table 3-1. The equations were solved with nominal values of Δt assumed. The Newton-Raphson method was used to compute solutions; three or four iterations were usually required before corrections became less than 0.5×10^{-5}. Initial estimates were the nominal multipliers' values.

No reason was found for the case where convergence did not occur. The tolerance on the corrections to successive iterates was increased to 10^{-3} but there was no convergence. The derivations and coding for this system of equations is being checked for errors.

Table 3-1. SOLUTIONS OF EQUATIONS FOR MULTIPLIERS

Nominal Multipliers	Computed Solutions
$\lambda_1 = 0.97371974$	$\lambda_1 = 0.98382817$
$\lambda_2 = 0.22774957$	$\lambda_2 = 0.17911486$
$\lambda_3 = -0.17809198 \times 10^{-2}$	$\lambda_3 = -0.43654483 \times 10^{-2}$
$\lambda_4 = 0.45610352 \times 10^{-2}$	$\lambda_4 = 0.42815572 \times 10^{-2}$

Radius condition represented by third-order series in Δt

Orthogonality condition represented by third-order series in Δt

Rendezvous condition represented by fifth-order series in Δt

$\Delta t = 170.34$ sec Nominal Trajectory AA-1

Nominal Multipliers	Computed Solutions
$\lambda_1 = 0.97371974$	$\lambda_1 = 0.96952223$
$\lambda_2 = 0.22774957$	$\lambda_2 = 0.24500336$
$\lambda_3 = -0.17809198 \times 10^{-2}$	$\lambda_3 = -0.17845442 \times 10^{-2}$
$\lambda_4 = 0.45610352 \times 10^{-2}$	$\lambda_4 = 0.48520438 \times 10^{-2}$

Radius condition represented by fourth-order series in Δt

Orthogonality condition represented by third-order series in Δt

Rendezvous condition represented by fifth-order series in Δt

$\Delta t = 170.34$ sec Nominal Trajectory AA-1

Nominal Multipliers	Computed Solutions
$\lambda_1 = 0.97371974$	Non-Convergence
$\lambda_2 = 0.22774957$	Non-Convergence
$\lambda_3 = -0.17809198 \times 10^{-2}$	Non-Convergence
$\lambda_4 = 0.45610352 \times 10^{-2}$	Non-Convergence

Radius condition represented by fifth-order series in Δt

Orthogonality condition represented by fourth-order series in Δt

Rendezvous condition represented by fifth-order series in Δt

$\Delta t = 170.34$ sec Nominal Trajectory AA-1

SECTION IV

OPTIMAL ASCENT TO ELLIPTICAL ORBIT

4.1 INTRODUCTION

A third guidance problem studied was that of optimal ascent to elliptical orbit. The approach to solution is the same as for the previously described problems except that the terminal boundary conditions are different. It is assumed that the orbit is described by its associated, specific energy and magnitude of angular momentum, and the direction of perigee with respect to an earth-centered coordinate system. These terminal conditions may be expressed through the following equations:

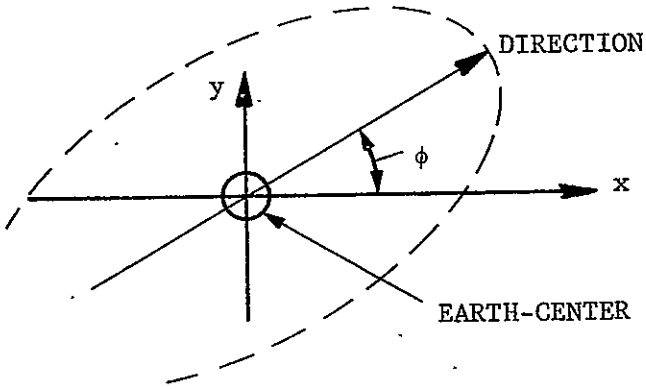
$$V_f^2 - 2\,\frac{GM}{R_f} - 2\,E_{CO} = 0 \qquad \text{(Energy)}$$

$$\left|\bar{R}_f \times \bar{V}_f\right| - \left|\bar{M}_{CO}\right| = 0 \qquad \text{(Momentum)}$$

$$\cos\phi + \left(\frac{x}{eR} + \frac{u(\bar{R}\cdot\bar{V}) - xV^2}{e\,GM}\right)_f = 0 \qquad \text{(Perigee)}$$

$$\left(\frac{GM}{R_f}\right)\bar{\lambda}_f \cdot \bar{R}_f + \dot{\bar{\lambda}}_f \cdot \bar{V}_f = 0 \qquad \text{(Transversality).}$$

The first two conditions determine the major axis and eccentricity of the orbit. The third condition is derived from "Hamilton's Integral" for the two-body problem; ϕ is the angle between the x-axis and the perigee direction. The coordinate system is shown in the following sketch.



As noted, the approach to solution for this problem is the same as for the two problems described previously. A nominal trajectory (i.e., a numerical solution for optimal ascent to elliptical orbit) was obtained. Data from this nominal case was then used to test, numerically, the analytical developments. Attempts were made to achieve correlation between the nominal case and values predicted by analytical approximations. At the time that these comparisons were made, two-dimensional nominal trajectories were not available. A nominal case was used that was very slightly three-dimensional; out-of-plane flight was only a few meters.

4.2 SOLUTION FOR FLIGHT TIME (Δt)

In the optimal ascent to circular orbit and rendezvous problems, the expression for Δt was obtained by inversion of one of the series expansions for a terminal condition. This expression could then be substituted into the other expansions and the explicit appearance of Δt would be eliminated. However, it may be desired to solve five equations in five unknowns if Δt is not eliminated.

If Δt is eliminated by substitution, one should choose the "best" series to invert for Δt. The "best" series is the one that requires the fewest terms to implicitly define Δt and also requires the fewest terms to invert. To determine which series would be the best candidate for inversion, all four series expansions were taken to third order and their coefficients were evaluated with nominal data. These third-degree polynomials in Δt were then solved for all of their roots. The third-order expansion for the terminal energy condition appears to define the most accurate values for Δt.

In addition, approximations for Δt and its powers, analogous to those in Section II, were attempted. Let the series expansion be

$$W = A_1 \Delta t + A_2 \Delta t^2 + A_3 \Delta t^3$$

where

$$W = \left\{ \frac{2E_{CO} - \overline{V} \cdot \overline{V} + 2\left(\frac{GM}{R}\right)}{2\left[\overline{V} \cdot \dot{\overline{V}} - \left(\frac{GM}{R}\right) \overline{R} \cdot \overline{V}\right]} \right\}$$

$$A_1 = 1$$

$$A_2 = \left\{ \frac{2\dot{\overline{V}} \cdot \dot{\overline{V}} + 2\overline{V} \cdot \ddot{\overline{V}} + 3\left(\frac{GM}{R}\right)\left(\overline{R} \cdot \overline{V}\right)^2 - \left(\frac{GM}{R^3}\right)\left(\overline{V} \cdot \overline{V} + \overline{R} \cdot \dot{\overline{V}}\right)}{2\overline{V} \cdot \dot{\overline{V}} - \left(\frac{GM}{R}\right) \overline{R} \cdot \overline{V}} \right\}$$

etc.

Assume that an estimate to Δt, defined as τ, is available. After following through the steps outlined in Section II, the following approximations are obtained:

$$\Delta t \approx \tau + dt_A$$

$$\Delta t^2 \simeq \tau^2 + 2\tau \, dt_A$$

$$\Delta t^3 \simeq \tau^3 + 3\tau^2 \, dt_A \quad ,$$

where

$$dt_A = \frac{W - \left(\tau + A_2\tau^2 + A_3\tau^3\right)}{1 + A_2\tau^2}$$

Figure 4-1 shows the result of using various values of τ to estimate Δt and its powers. While the Δt values predicted by the formula above are not close to the true Δt values, they are close to the actual root of the polynomial that is closest to the true Δt value. If the series expansions of the terminal energy condition were taken to the fourth or fifth order, and then inverted, the approximations would probably be closer to the true values.

However, because the approximations for Δt are rather complicated in this problem, it is advisable to consider solving for Δt simultaneously with the multipliers. There was not enough time available to follow through on this possibility during the term of this contract.

4.3 SOLUTION FOR MULTIPLIERS

In solving for the multipliers, it was assumed that Δt would be eliminated by substitution of the inverted series for the energy condition. Thus, the series for the perigee, transversality, and momentum, together with the scaling condition on the multipliers, would be used to derive a system of algebraic equations that could be solved for the initial values of the multipliers. These algebraic equations were derived by substituting the differential equations' derivatives into the third-order expansions.

Attempts to solve, numerically, this system of four equations in four unknowns failed. Derivations and program coding were checked for errors but none were apparent. The expansion of the transversality condition to third order has a very large remainder term when evaluated with nominal data. It is likely that if this series is extended to higher orders, a solution for the multipliers can be achieved. Again, there was insufficient time available to pursue this problem during the term of this contract.

Figure 4-1 shows the result of using various values of τ to estimate Δt and its powers. While the Δt values predicted by the formula above are not close to the true Δt values, they are close to the actual root of the polynomial that is closest to the true Δt value. If the series expansions of the terminal energy condition were taken to the fourth or fifth order, and then inverted, the approximations would probably be closer to the true values.

However, because the approximations for Δt are rather complicated in this problem, it is advisable to consider solving for Δt simultaneously with the multipliers. There was not enough time available to follow through on this possibility during the term of this contract.

4.3 SOLUTION FOR MULTIPLIERS

In solving for the multipliers, it was assumed that Δt would be eliminated by substitution of the inverted series for the energy condition. Thus, the series for the perigee, transversality, and momentum, together with the scaling condition on the multipliers, would be used to derive a system of algebraic equations that could be solved for the initial values of the multipliers. These algebraic equations were derived by substituting the differential equations' derivatives into the third-order expansions.

Attempts to solve, numerically, this system of four equations in four unknowns failed. Derivations and program coding were checked for errors but none were apparent. The expansion of the transversality condition to third order has a very large remainder term when evaluated with nominal data. It is likely that if this series is extended to higher orders, a solution for the multipliers can be achieved. Again, there was insufficient time available to pursue this problem during the term of this contract.

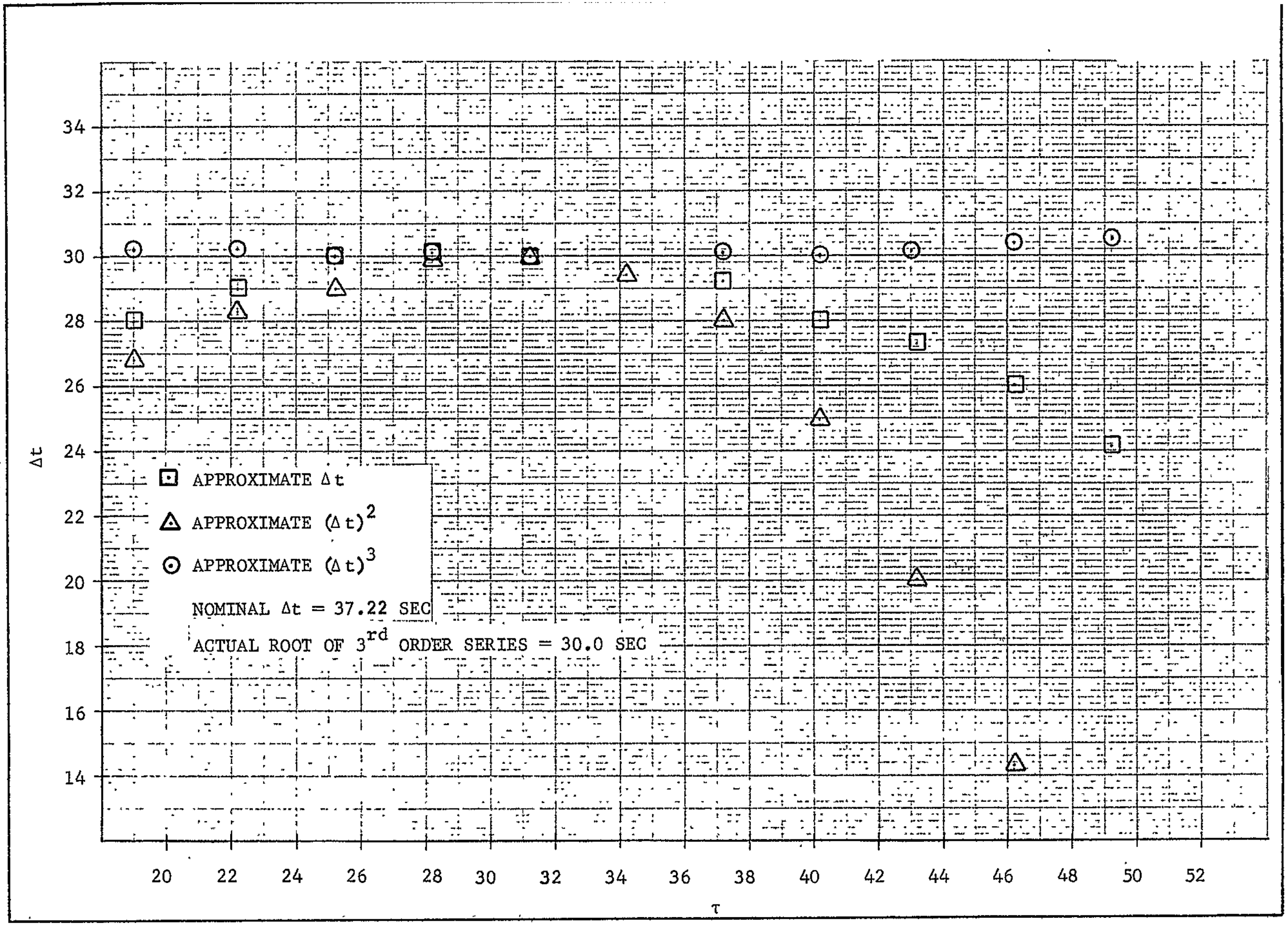


Figure 4-1. APPROXIMATIONS TO Δt FOR DIFFERENT VALUES OF τ

TR-792-8-283

SECTION V

SYMBOLIC MANIPULATION OF POLYNOMIALS

5.1 BACKGROUND

An outstanding characteristic of the work described in this report is the amount of algebraic manipulations required. These manipulations can be put into two classes: (1) Differentiation, and (2) Substitution, multiplication, and simplification of polynomial-like expressions.

There are presently available a number of computer languages that are more or less suitable for handling these kinds of symbolic mathematical problems. One language that appears to be strongly oriented toward function differentiation is FORMAC. It, like the other symbol-manipulation languages known to the authors, suffers from an inability to exploit off-line on "non-core" storage.

By the time the significantly difficult parts of our analytical developments were reached, the storage facilities of FORMAC had been exhausted. Attempts were made to segment problems into portions that could be handled by FORMAC. But it was soon evident that the bookkeeping problems associated with this approach offset the advantages. This is not to say that FORMAC or a similar language is without value, but that for the problems we dealt with it was of relatively small utility.

It was apparent that most of our symbolic computations involved polynomial-like structures; e.g., power series or determinant expansions. Most of the operations required with these expressions involved multiplication as the most difficult step; e.g., substitution of polynomials. These are also the kinds of operations that the symbol-manipulating languages fail to perform for polynomials that have a large number of terms.

Because of these difficulties, it was decided to develop a computer program, written in a general purpose language, that would accomplish at least the multiplication of polynomials.

5.2 THE MULPO PROGRAM

The MULPO Program is designed to multiply polynomials, in one to six variables, which have literal coefficients. An additional feature is the ability to add polynomials of the same type as it can multiply. Thus, the program can perform substitution of polynomials into other polynomials. It is written entirely in FORTRAN IV, and is intended for use on the IBM 7094. However, it can be run, with minor changes, on another machine with a comparable FORTRAN IV translator and enough tape units. A users guide for MULPO is given in reference 3.

At one point in its development, MULPO was written in ALGOL 60 for the Burroughs B5500 computer. The reason for this was to obtain faster execution speed. Due to the widespread use of FORTRAN, however, it was converted to that language. For all but unusually long problems or volumes of production work, the relative inefficiency should not be noticeable.

As a test of its ability to handle large-scale problems, the program was asked to multiply four polynomials together; then add this result to the product of three polynomials; then add this result to the product of two other polynomials. Each polynomial was in terms of four variables and consisted of about ten terms each. The result was a polynomial of over 11,000 terms; and its like powers and coefficients were combined.

In addition to studies such as reported here, a program like this probably has numerous other applications; for example, in perturbation techniques in celestial mechanics.

5.3 DIRECTIONS FOR READING OUTPUT FROM MULPO

Consider the following problem:

Given:

$$P_o = \sum A_{ijkl} \lambda_1^i \lambda_2^j \lambda_3^k \lambda_4^l$$

$$P_1 = \sum B_{ijkl} \lambda_1^i \lambda_2^j \lambda_3^k \lambda_4^l$$

$$P_2 = \sum C_{ijkl} \lambda_1^i \lambda_2^j \lambda_3^k \lambda_4^l$$

$$P_3 = \sum D_{ijkl} \lambda_1^i \lambda_2^j \lambda_3^k \lambda_4^l$$

$$Q_1 = \sum X_{ijkl} \lambda_1^i \lambda_2^j \lambda_3^k \lambda_4^l$$

$$Q_2 = \sum Y_{ijkl} \lambda_1^i \lambda_2^j \lambda_3^k \lambda_4^l$$

$$Q_3 = \sum Z_{ijkl} \lambda_1^i \lambda_2^j \lambda_3^k \lambda_4^l$$

where $Y_{ijkl} = K_1 X_{ijkl}$

$Z_{ijkl} = K_2 X_{ijkl}$

Form

$$S = \sum S_{ijkl} \lambda_1^i \lambda_2^j \lambda_3^k \lambda_4^l = P_o + P_1 Q_1 + P_2 Q_2 + P_3 Q_3$$

and "simplify" S by collecting like combinations of powers resulting from multiplications, and by combining like coefficients when they occur as sums in a like combination of powers. For this example, an S-polynomial term will generally have a multi-term coefficient made up of a term from P_o (A-coefficient) a term from the product of P_1Q_1 (B-coefficients times X-coefficients), a term from P_2Q_2 (C-coefficients times Y-coefficients), and a term from P_3Q_3 (D-coefficients times Z-coefficients).

Referring to Table 2-2, it can be seen that the integers directly under L1, L2, etc are the powers of the lambdas in the S-polynomial. (Note that the output is arranged so that the polynomial is ordered according to increasing powers.) The integer on the next line is the numerical multiplier of the literal coefficient on the next line down.

The symbol for the coefficient comes first, then the next four integers represent its subscript. The fifth integer is the exponent on the coefficient. (The asterisks should be ignored when they appear.) When the remainder of the line is blank, interpret this as a plus sign and go to the next line. If only <u>one</u> more term appears (i.e., five integers), then these integers represent the four subscripts and the exponent of the coefficient in a Q-polynomial that multiplies the term immediately to the left.

Which Q-polynomial is involved depends on the coefficient symbol at the start of the line. The coefficients of the Q_1 polynomial always multiply the B-coefficient, the Q_2 polynomial's coefficient the C-coefficients, etc.

If more than three terms are on one line, the last term is a coefficient of a Q-polynomial. The preceding coefficients are, as before, identified by the alphabetic symbol at the far left. In some cases enough coefficients are combined to cause the program to carry over the remainder of a coefficient to the next line. In such cases the appropriate alphabetic symbol reappears at the beginning of the next line.

Actually, reading the output is far simpler than describing how to do so. We can translate the first four terms of Table 2-2 into conventional notation as:

Coefficient of $\lambda_1^0 \lambda_2^0 \lambda_3^0 \lambda_4^0$

A_{0000} +
B_{0000} X_{0000} +
C_{0000} Y_{0000} +
D_{0000} Z_{0000}

Coefficient of $\lambda_1^0 \lambda_2^0 \lambda_3^0 \lambda_4^1$

C_{0001} Z_{0000}

Coefficient of $\lambda_1^0 \lambda_2^0 \lambda_3^1 \lambda_4^0$

D_{0010} Z_{0000}

Coefficient of $\lambda_1^0 \lambda_2^1 \lambda_3^0 \lambda_4^0$

B_{0000} X_{0100} +
C_{0000} Y_{0100} +
C_{0100} Y_{0000} +
D_{0000} Z_{0100} +
D_{0100} Z_{0000}

SECTION VI
CONCLUSIONS AND RECOMMENDATIONS

Based on the studies conducted under this contract, several conclusions and recommendations can be made. The first part of this section deals with the study described in this report. The second part is concerned with recommended modifications to the approach described herein and possible alternative approaches to obtain explicit guidance formulas.

A major accomplishment has been the successful automation, on digital computers, of the manipulation of polynomials that have literal coefficients. This computer program has an advantage over other, general purpose symbol manipulating languages because there are no practical limitations on computer storage. In conjunction with a language such as FORMAC, it should give an analyst the capability of multiplying or substituting power series or polynomials of virtually any number of terms. The program was designed to accommodate polynomials or power series involving as many as six variables, should the need arise in the future. It was also coded in a general purpose language (FORTRAN IV) that is commonly used and should be available in the future. It is also possible, but not easy, to alter the program to perform partial differentiation of multivariable power series or polynomials.

Much of the effort during this study was directed toward simplification of the analytical developments. The term "simplification" is taken to mean the dropping of terms in formulas for the guidance functions when these terms do not significantly influence the answers, or approximating complicated formulas with simpler ones. The simplification process was rather successful in the formulas for Δt; particularly for ascent to circular orbit and rendezvous in circular orbit. However, no simplifications in the solution for the multipliers or guidance function beyond those previously reported were achieved. A further test of these simplifications was made by perturbing the initial values of the state variables on a nominal trajectory. Simplifications that were valid on the nominal trajectory were also valid on neighboring trajectories.

Application of the Successive Substitution method to solve the equations in the multipliers is recommended only when the coefficients of the system of equations are numbers. Application to a system having literal coefficients results in an impractical number of terms. If approximate values of the initial multipliers are assumed known, the method of inversion should be tried. This approach to solution of the equations in the multipliers is analogous to that used in Section II for solution for Δt. Such an approach could allow for a considerable amount of simplification as was found in the approximations for Δt.

No solution was obtained for the multipliers in the problem of optimal ascent to elliptical orbit. This is believed to be due to the insufficient order of the series expansion for the transversality condition. It is recommended that this problem be studied further, series orders extended, and solutions for the multipliers be attempted. Because of the complexity of the formula for Δt that is got by inversion of the energy condition expansion, consideration should be given to solving for Δt simultaneously with the multipliers.

SECTION VII

BOLZA PROBLEMS WITH END ORBITS

The work discussed in reference 2 pertaining to Bolza problems is not closely related to the topics discussed in other parts of this Summary Report. However to keep the reader informed on all aspects of the contract efforts, an abstract of reference 2 is provided in the following paragraphs.

It is well known that many of the problems of space vehicle guidance can be formulated in such a way as to be susceptible to the methods of the calculus of variations. The problems considered fall roughly within the class of problems referred to as problems of Bolza.

Many space missions are formulated in such a way that considerable further analysis is needed to bring the problem within the format of the Bolza problem, as regards the end-point constraints. It is usually the case that conditions are imposed on the orbits determined by the end points of the burn trajectory, but the translation of these conditions into conditions applying directly to the end point values of the states and time requires that the orbital motion be expressible as a function of the end-point conditions and time, and that the conditions on that orbital motion then be invertible to yield conditions on the end-point values. Both of these problems can be quite profound, so that it is not often possible to give analytic expressions for the end-point constraints, although these are assumed available in the formulation of the problem of Bolza.

We therefore formulate a modified Bolza problem, called a Bolza problem with end orbits. The objective is to define the problem in terms of that information which actually is available, and to obtain conditions which apply to that available information. Thus in the new problem formulation, there are given differential equations (analogous to the gravity equations) to which the end points of any admissible curve determine particular solutions, called the end orbits of that curve. Other conditions are given, which apply to the end orbits (not to the end points), called end-orbit constraints, or simply orbital constraints. The transversality condition is then deduced in such a form as to apply directly to the differential equations of the end orbits and the orbital constraints. In particular, it is not required that the differential equations of the end orbits be solved.

SECTION VIII

REFERENCES

1. Gilchrist, C. A., et al. "An Analytical Approach to Solution of Two-Point Boundary Condition Problems in Optimal Guidance," Northrop-Huntsville, TM-292-6-038, June 1966.

2. Silber, R., "Bolza Problems With End Orbits: Studies on the Application of the Theory of Bolza's Problem to Space Missions," Northrop-Huntsville, TM-792-7-147, March 1967.

3. Thompson, M. L. and R. C. Armstrong, "Automated Analytic Procedures for Obtaining Optimal Guidance Functions," Interim Report, Nortronics-Huntsville, TR-792-7-265, October 1967.

APPENDIX A

TIME DERIVATIVES OF DIFFERENTIAL EQUATIONS

$$\dot{\overline{V}} = \alpha_1 \overline{\lambda} - \beta_1 \overline{R}$$

$$\ddot{\overline{V}} = \alpha_2 \overline{\lambda} + \alpha_1 \dot{\overline{\lambda}} - \alpha_1 (\overline{\lambda} \cdot \dot{\overline{\lambda}}) \overline{\lambda} - \beta_2 \overline{R} - \beta_1 \overline{V}$$

$$\dddot{\overline{V}} = \alpha_3 \overline{\lambda} + 2\alpha_2 \dot{\overline{\lambda}} - 3\alpha_2 (\overline{\lambda} \cdot \dot{\overline{\lambda}}) \overline{\lambda} + \Delta_1 (\overline{\lambda} \cdot \overline{R})$$

$$-2\alpha_1 (\overline{\lambda} \cdot \dot{\overline{\lambda}}) \dot{\overline{\lambda}} - \alpha_1 (\dot{\overline{\lambda}} \cdot \dot{\overline{\lambda}}) \overline{\lambda} + \alpha_1 (\overline{\lambda} \cdot \dot{\overline{\lambda}})^2 \overline{\lambda} + \Delta_2$$

$$\ddot{\overline{\lambda}} = \beta_1 \overline{\lambda} + \gamma (\overline{\lambda} \cdot \overline{R}) \overline{R}$$

$$\dddot{\overline{\lambda}} = \beta_2 \overline{\lambda} + \beta_1 \dot{\overline{\lambda}} + \left[\rho (\overline{\lambda} \cdot \overline{R}) + \gamma (\overline{\lambda} \cdot \overline{R} + \overline{\lambda} \cdot \overline{V})\right] \overline{R} + \gamma (\overline{\lambda} \cdot \overline{R}) \overline{V}$$

where

$$\alpha_1 = \frac{F}{m}, \quad \alpha_2 = -\frac{F}{m}\frac{\dot{m}}{m}, \quad \alpha_3 = \frac{F}{m}\left(\frac{\dot{m}}{m}\right)^2$$

$$\beta_1 = \frac{GM}{R^3}, \quad \beta_2 = -3\left(\frac{GM}{R^5}\right)(\overline{R} \cdot \overline{V})$$

$$\Delta_1 = (\alpha_1 \gamma + \alpha_1 \beta_1) \overline{R}, \quad \gamma = \frac{\beta_2}{(\overline{R} \cdot \overline{V})}, \rho = 15\left(\frac{GM}{R^7}\right)(\overline{R} \cdot \overline{V})$$

$$\Delta_2 = \left[\beta_1^2 - \beta_1^2 R^2 + \beta_2 (\overline{R} \cdot \overline{V}) + \beta_1 (\overline{V} \cdot \overline{V})\right] \overline{R} - 2\beta_2 \overline{V}$$

All variables are evaluated at $t = t_o$.

APPENDIX B

INVERSION OF SERIES IN SEVERAL VARIABLES

Suppose that we are given the two equations in two variables.

$$F_1(x_1,x_2) = 0$$

$$F_2(x_1,x_2) = 0.$$

We wish to solve them for roots $\bar{x}_1$ and $\bar{x}_2$. Let the functions F_1 and F_2 be expanded about a point $\tilde{x}_1$ and $\tilde{x}_2$ which is presumably near $\bar{x}_1$ and $\bar{x}_2$.

$$F_1(\bar{x}_1,\bar{x}_2) = F_1(\tilde{x}_1,\tilde{x}_2) + \frac{\partial F_1}{\partial x_1}(\bar{x}_1 - \tilde{x}_1) + \frac{\partial F_1}{\partial x_2}(\bar{x}_2 - \tilde{x}_2) + \cdots = 0$$

and similarly for F_2. These expansions are rewritten as

$$\alpha_{00} = \alpha_{10} X_1 + \alpha_{01} X_2 + \alpha_{20} X_1^2 + \cdots$$

$$\beta_{00} = \beta_{10} X_1 + \beta_{01} X_2 + \beta_{20} X_1^2 + \cdots$$

where

$$X_1 = (\bar{x}_1 - \tilde{x}_1)$$

$$X_2 = (\bar{x}_2 - \tilde{x}_2)$$

$$-\alpha_{00} = F_1(\tilde{x}_1,\tilde{x}_2)$$

$$-\beta_{00} = F_2(\tilde{x}_1,\tilde{x}_2).$$

We assume that the inverse may be represented as

$$X_1 = A_{10}\,\alpha_{00} + A_{01}\,\beta_{00} + A_{20}\,\alpha_{00}^2 + A_{11}\,\alpha_{00}\,\beta_{00} + A_{02}\,\beta_{00}^2 + \cdots$$

$$X_2 = B_{10}\,\alpha_{00} + B_{01}\,\beta_{00} + B_{20}\,\alpha_{00}^2 + B_{11}\,\alpha_{00}\,\beta_{00} + B_{02}\,\beta_{00}^2 + \cdots$$

It is now necessary to determine the coefficients of the inverse series in terms of the coefficients in the original series. The derivations will be simpler if we write the inverse as

$$X_1 = Z_1 + Z_2 + Z_3 + \cdots$$

$$X_2 = W_1 + W_2 + W_3 + \cdots,$$

where Z_n and W_n stand for all terms of degree n in the α's and β's.

The assumed form for the inverse is substituted into the original series and the method of undetermined coefficients is used to determine the Z_n and W_n. For the first-degree terms we have

$$\begin{vmatrix} Z_1 \\ W_1 \end{vmatrix} = \begin{Vmatrix} \alpha_{10} & \alpha_{01} \\ \beta_{10} & \beta_{01} \end{Vmatrix}^{-1} \begin{vmatrix} \alpha_{00} \\ \beta_{00} \end{vmatrix}$$

For the second-degree terms we have

$$\begin{vmatrix} Z_2 \\ W_2 \end{vmatrix} = \begin{Vmatrix} \alpha_{10} & \alpha_{01} \\ \beta_{10} & \beta_{01} \end{Vmatrix}^{-1} \begin{vmatrix} K_1 \\ J_1 \end{vmatrix}$$

where

$$-K_1 = \alpha_{20}\, Z_1^2 + \alpha_{11}\, Z_1\, W_1 + \alpha_{02}\, W_1^2$$

$$-J_1 = \beta_{20}\, Z_1^2 + \beta_{11}\, Z_1\, W_1 + \beta_{02}\, W_1^2$$

In general,

$$\begin{vmatrix} X_1 \\ X_2 \end{vmatrix} = \begin{vmatrix} Z_1 \\ W_1 \end{vmatrix} + \begin{vmatrix} Z_2 \\ W_2 \end{vmatrix} + \cdots$$

APPENDIX C

SIMPLIFIED SERIES FOR Δt FOR ASCENT TO CIRCULAR ORBIT AND RENDEZVOUS PROBLEMS

$$\Delta t = \beta_{0000} + \beta_{1000}\lambda_1 + \beta_{0100}\lambda_2 + \beta_{1101}\lambda_1\lambda_2\lambda_4 + \beta_{0201}\lambda_2^2\lambda_4$$
$$+ \beta_{0120}\lambda_2\lambda_3^2 + \beta_{1002}\lambda_1\lambda_4^2 + \beta_{3000}\lambda_1^3 + \beta_{0300}\lambda_2^3$$
$$+ \beta_{2111}\lambda_1^2\lambda_2\lambda_3\lambda_4 + \beta_{1202}\lambda_1\lambda_2^2\lambda_4^2 + \beta_{2120}\lambda_1^2\lambda_2\lambda_3^2$$
$$+ \beta_{1211}\lambda_1\lambda_2^2\lambda_3\lambda_4 + \beta_{1011}\lambda_1\lambda_3\lambda_4 + \beta_{0111}\lambda_2\lambda_3\lambda_4$$
$$+ \beta_{1220}\lambda_1\lambda_2^2\lambda_3^2 + \beta_{2102}\lambda_1^2\lambda_2\lambda_4^2 + \beta_{0210}\lambda_2^2\lambda_3 + \beta_{2001}\lambda_1^2\lambda_4$$

where

$$\beta_{0000} = [\beta_1 V^2 + \beta_2(\overline{R}\cdot\overline{V}) - \alpha_1^2 - (\beta_1 R)^2]\ \tau^2/2\delta + \tau + Z/(1-2C)$$
$$-[3\beta_2 V^2 - 8\beta_1{}^2(\overline{R}\cdot\overline{V}) + \rho(\overline{R}\cdot\overline{V})^2 + 3\alpha_1\alpha_2]\ \tau^3/6\delta$$

$$\beta_{1000} = -[\alpha_2 u - 2\alpha_1\beta_1 x]\ \tau^2/2\delta$$
$$-[\alpha_3 u - 3\alpha_2\beta_1 x - 3\beta_2\alpha_1 x - 4\alpha_1\beta_1 u - \alpha_1\gamma y^2 u - 2\alpha_1\gamma xyv]\ \tau^3/6\delta$$

$$\beta_{0100} = -[\alpha_2 v - 2\alpha_1\beta_1 y]\ \tau^2/2\delta$$
$$-[\alpha_3 v - 3\alpha_2\beta_1 y - 3\beta_2\alpha_1 y - 4\alpha_1\beta_1 v - \alpha_1\gamma x^2 v - 2\alpha_1\gamma xyu]\ \tau^3/6\delta$$

$$\beta_{1101} = -[\alpha_1 u]\tau^2/2\delta - [2\alpha_2 u - 3\alpha_1\beta_1 x]\ \tau^3/6\delta$$

$$\beta_{0201} = -[\alpha_1 v]\tau^2/2\delta - [2\alpha_2 v - 3\alpha_1\beta_1 y]\ \tau^3/6\delta$$

$$\beta_{0120} = [\alpha_1 v]\tau^3/6\delta$$

$$\beta_{1002} = [\alpha_1 u]\tau^3/6\delta$$

$$\beta_{3000} = [\alpha_1\gamma ux^2 - \alpha_1\gamma uy^2 - 2\alpha_1\gamma xyv]\tau^3/6\delta$$

$$\beta_{0300} = [\alpha_1 \gamma v y^2 - \alpha_1 \gamma v x^2 - 2\alpha_1 \gamma x y u]\tau^3/6\delta$$

$$\beta_{2111} = - [6\alpha_1 u]\tau^3/6\delta$$

$$\beta_{1202} = - [3\alpha_1 u]\tau^3/6\delta$$

$$\beta_{2120} = - [3\alpha_1 v]\tau^3/6\delta$$

$$\beta_{1211} = - [7\alpha_1 v]\tau^3/6\delta$$

$$\beta_{1011} = [2\alpha_1 v]\tau^3/6\delta$$

$$\beta_{0111} = [2\alpha_1 u]\tau^3/6\delta$$

$$\beta_{1220} = [3\alpha_1 u]\tau^3/6\delta$$

$$\beta_{2102} = [3\alpha_1 v]\tau^3/6\delta$$

$$\beta_{0210} = [\alpha_1 u \ \tau^2/2\delta + 2\alpha_2 u - 3\beta_1 \alpha_1 x] \ \tau^3/6\delta$$

$$\beta_{2001} = [\alpha_1 v \ \tau^3/2\delta + 2\alpha_2 v - 3\beta_1 \alpha_1 y] \ \tau^3/6\delta$$

where

$$\beta_1 = GM/R^3$$

$$\beta_2 = - 3(GM/R^3)(\bar{R}\cdot\bar{V})$$

$$\gamma = -3GM/R^5$$

$$\rho = 15 \ (GM/R^7)(\bar{R}\cdot\bar{V})$$

$$\delta = (\bar{V}\cdot\dot{\bar{V}}) + (\dot{\bar{V}}\cdot\dot{\bar{V}})$$

$$\alpha_1 = F/m$$

$$\alpha_2 = -F/m \ (\dot{m}/m)$$

$$C = \dot{\bar{V}} \cdot \dot{\bar{V}}/2\bar{V} \cdot \dot{\bar{V}}$$

τ = the estimated value of Δt.